Towards Exploring Fundamental Limits of System-Specific Cryptanalysis Within Limited Attack Classes: Application to ABSG

Yücel Altuğ, M. Kıvanç Mıhçak *Member*

Abstract

A new approach on cryptanalysis is proposed where the goal is to explore the fundamental limits of a specific class of attacks against a particular cryptosystem. As a first step, the approach is applied on ABSG, which is an LFSR-based stream cipher where irregular decimation techniques are utilized. Consequently, under some mild assumptions, which are common in cryptanalysis, the tight lower bounds on the algorithmic complexity of successful Query-Based Key-Recovery attacks are derived for two different setups of practical interest. The proofs rely on the concept of "typicality" of information theory.

I. Introduction

In this paper, we introduce a (to the best of our knowledge) novel approach to cryptanalysis. In our approach, the focus is jointly on a particular cryptosystem and a specific (sufficiently broad) class of attacks of interest at the same time. Then, under some mild conditions, the goal is to derive the *achievable fundamental performance limit* for the attacks within the considered class of interest against the cryptosystem at hand. The aforementioned limit should be "achievable", in the sense that it is necessary to provide an explicit attack construction of which performance coincides with the derived limit. Furthermore, the aforementioned limit should also necessarily be "fundamental", in the sense that within the considered specific class, there does not exist any attack of which performance is superior to the derived limit.

Our proposed approach contrasts with the trend in conventional cryptanalysis, which can be outlined in two categories. In the first category, the focus is on the construction of a generic attack, which should be applicable (subject to slight modifications) to most cryptosystems; common examples include time-memory tradeoff attacks [1], [2], correlation attacks [3], [4], algebraic attacks [5], [6] and alike. The second category is conceptually on the opposite side of the spectrum. Here, given a particular cryptosystem, the focus is on the construction of a potentially-specialized attack, which is "tailored" specifically against the system at hand; hence, the resulting attack is not applicable to a broader class of cryptosystems in general. Although the approaches pursued in the aforementioned two attack categories are radically different, it is interesting to note that, for both of them the underlying fundamental goal is the same, which can be summarized as providing a "design advice" to the cryptosystem designer. In practice, at first, the cryptosystem designer is expected to test his/her proposed system against generic attacks (first category); thus, such attacks serve as a benchmark for the community of cryptosystem designers. Next, the cryptanalyst tests a proposed cryptosystem via constructing a cryptosystem-specific attack algorithm (second category). Both categories

The authors are with the Electrical and Electronic Engineering Department of Boğaziçi University, Istanbul, 34342, Turkey (e-mail: yucel.altug@boun.edu.tr, kivanc.mihcak@boun.edu.tr)

Y. Altuğ is partially supported by TÜBİTAK Career Award no. 106E117; M. K. Mıhçak is partially supported by TÜBİTAK Career Award no. 106E117 and TÜBA-GEBIP Award.

have been shown to be extremely valuable in practice since the first one provides a "unified approach" to cryptanalysis via providing some generic attack algorithms and the second one specifically tests the security of the considered cryptosystem and consequently yields its potential weaknesses. On the other hand, both categories of the conventional approach in cryptanalysis lack to provide fundamental performance bounds, i.e., the question of "what is the best that can be done?" goes unanswered. The main reason is that, for the first category, finding out a fundamental performance bound necessarily requires considering all possible cryptosystems, which is infeasible in practice; within the second category, providing a fundamental performance bound necessarily requires "describing" all possible cryptanalytic propositions (in a computational sense) and quantifying the resulting performances, which is again infeasible in practice.

In our proposed approach, we aim to derive "the best possible performance bound" [1] in a reasonably-confined setup. Intuitively, we "merge" the first and the second categories of the conventional cryptanalytic approach; we jointly focus on *both* a particular cryptosystem *and* a specific class of attacks, and subsequently aim to analytically quantify the fundamental, achievable performance bounds, i.e., specifically for a given cryptosystem, our goal is to find the achievable lower-bound on the complexity of a proposed class of attacks, under a set of mild assumptions. The main impact of this approach is that, it aims to provide an advice for the cryptanalyst, instead of the cryptosystem designer, in contrast with the conventional approach. If this resulting advice is "positive" (i.e., the fundamental achievable performance bound is of polynomial complexity), then the weakness of the analyzed cryptosystem is guaranteed (which can also be achieved via pursuing the second category of the conventional cryptanalytic approach). However, more interestingly, if the resulting advice is "negative" (i.e., the fundamental achievable performance bound is of exponential complexity), then the considered class of attacks is *guaranteed* to be useless, which, in turn, directs a cryptanalyst to consider different classes of attacks, instead of experimenting with various attacks from the considered class via a (possibly educated) trial-and-error approach. Thus, the negative advice case (for which this paper serves an exemplary purpose) constitutes the fundamental value of our approach. We believe that our efforts can be viewed as a contribution towards the goal of enhancing cryptanalytic approaches via incorporating a structural and procedural methodology.

In order to illustrate our approach, in this paper we consider a class of Query-Based Key-Recovery attacks (of which precise definition is given in Sec. III-B) targeted towards ABSG [7], which is an LFSR(linear feedback shift register)-based stream cipher that uses irregular decimation techniques. Recall that, within the class of stream ciphers, the usage of LFSR is an attractive choice due to the implementation efficiency and favorable statistical properties of the LFSR output; however, security of LFSR-based stream ciphers is contingent upon applying additional non-linearities per the linear nature of LFSR [8]. An approach, which aims to achieve this task, is to use irregular decimation techniques to the LFSR output [7], [9], [10], [11].

[1]Note that, this approach is analogous to providing both achievability and converse proofs in classical information-theory problems. This connection will further be clarified throughout the paper.

The motivation lying behind the development of this approach is to render most conventional attacks useless (such as algebraic attacks). Shrinking [10] and self-shrinking generators (SSG) [11] are two important examples of this approach. In particular, in the literature SSG is well-known to be a very efficient algorithm and it has been shown to possess favorable security properties [12], [13], [14]. The bit-search generator (BSG) [9] and its variant ABSG [7] are newer algorithms, which also use irregular decimation techniques. In [15], it has been shown that the efficiency (output rate) of ABSG is superior to that of SSG and the security level of ABSG is at least the same level provided by SSG under a broad class of attacks. A detailed analysis of the statistical properties of ABSG and BSG algorithms has recently been presented in [16]. Since ABSG has been shown to be a state-of-the-art cryptosystem, in our developments we focus on it under a reasonable class of attacks and subsequently provide "negative advices" for the cryptanalyst in various setups of interest. Next, we summarize our main results.

Main Results: Our contributions, which have been derived under a set of mild assumptions (specified in Sec. III-A), are as follows:

- We show that breaking ABSG algorithm is equivalent to "guessing" a sequence of random variables, which are i.i.d. (independent identically distributed) with geometric distribution of parameter $1/2$ using complexity theoretic notions (Theorem 3.1).
- In order to solve the problem mentioned in the previous item, we formulate a sufficiently broad class of attacks, termed as "Query-Based Key-Recovery attacks", which are quite generic by construction, and hence applicable for cryptanalysis for a wide range of cryptosystems (Definition 3.3).
- Within the class of attacks mentioned in the previous item, first we concentrate on a practically-meaningful subset of them (termed "Exhaustive-Search Type Query-Based Key-Recovery attacks") (Sec. IV); we derive a fundamental lower bound on the complexity of any successful attack in this subset (Theorem 4.2); this lower bound is proven to be achievable to the first order in the exponent (Theorem 4.1).
- We consider the set of all Query-Based Key-Recovery attacks (Sec. V); we derive a fundamental lower bound on the complexity of any successful attack within this set (Theorem 5.2), followed by stating the proof of the achievability result (to the first order in the exponent) using the "most probable choice" attack given in [7] (Theorem 5.1).

Organization of the Paper: In Section II, we present the notation used in the paper and recall the definition of ABSG. Section III provides the assumptions we have employed throughout the paper, the problem formulation and the definition of "Query-Based Key-Recovery" (QuBaR) attacks. In Section IV, we derive a tight (to the first order in the exponent) lower bound on the complexity of exhaustive-search type QuBaR attacks. In Section V, we derive a tight lower bound on the complexity of any QuBaR attack. We conclude with discussions given in Section VI.

II. Notation and Background

A. Notation

Boldface letters denote vectors; regular letters with subscripts denote individual elements of vectors. Furthermore, capital letters represent random variables and lowercase letters denote individual realizations of the corresponding random variable. The sequence of $\{a_1, a_2, \ldots, a_N\}$ is compactly represented by $\mathbf{a}_1^N$. Given $x \in \{0,1\}$, $\bar{x}$ denotes the binary complement of x. The abbreviations "i.i.d.", "p.m.f." and "w.l.o.g." are shorthands for the terms "independent identically distributed", "probability mass function" and "without loss of generality", respectively. Throughout the paper, all logarithms are base-2 unless otherwise specified. Given a discrete random variable X with the corresponding p.m.f. $p(\cdot)$, defined on the alphabet $\mathcal{X}$, its entropy (in bits) is $H(X) \stackrel{\triangle}{=} -\sum_{x\in\mathcal{X}} p(x)\log p(x)$. In the sequel, we say that "a_n and b_n are equal to the first order in the exponent" provided that $\lim_{n\to\infty} \frac{1}{n}\log\frac{a_n}{b_n} = 0$, which is denoted by $a_n \doteq b_n$ in our notation.

B. Background

Throughout this paper, we use the notation that was introduced in [16].

Definition 2.1: Given an infinite length binary sequence $\mathbf{x} = \{x_n\}_{n=1}^{\infty}$ which is an input to the ABSG algorithm, we define

- $\mathbf{y} \triangleq \mathcal{A}(\mathbf{x})$, where the sequence $\mathbf{y}$ represents the internal state of the ABSG algorithm and $y_i \in \{\varnothing, 0, 1\}$, $1 \le i < \infty$. The action of algorithm $\mathcal{A}$ is defined via the recursive mapping $\mathcal{M}$:

$$y_i = \mathcal{M}(y_{i-1}, x_i), \quad 1 \le i < \infty,$$

with the initial condition $y_0 = \varnothing$. The mapping $\mathcal{M}$ is defined in Table I .

TABLE I
TRANSITION TABLE OF ALGORITHM $\mathcal{A}$

$y_{i-1}\backslash x_i$	0	1
$\varnothing$	0	1
0	$\varnothing$	0
1	1	$\varnothing$

- $\mathbf{z} \triangleq \mathcal{B}(\mathbf{y})$, where the sequence $\mathbf{z}$ represents the output of the ABSG algorithm, such that the action of the algorithm $\mathcal{B}$ is given as follows:

$$z_j = \begin{cases} y_{i-1}, & \text{if } y_i = \varnothing \text{ and } y_{i-2} = \varnothing, \\ \bar{y}_{i-1}, & \text{if } y_i = \varnothing \text{ and } y_{i-2} \neq \varnothing, \end{cases}$$

where $j \le i$ and $i, j \in \mathbb{Z}^+$.

From Definition 2.1, we clearly deduce that the ABSG algorithm produces an output bit (z_j denoting the j-th output bit) if and only if the value of the corresponding internal state variable (y_i denoting the value of the internal state variable at time i) is $\varnothing$. The fact that $y_i \neq \varnothing$ for all i is the reason of the mismatch between the input sequence indices (which are the same as the indices of the internal state variables) and the output sequence indices.

III. Problem Setup and Formulation

A. Assumptions and Preliminaries

Throughout this paper, we consider the type of attacks, in which retrieving L (where L is the degree of the feedback polynomial of the generating LFSR) linear equations in terms of $\mathbf{x}_1^M$ is aimed. This type of attacks correspond to *key recovery attacks* to ABSG (assuming that the feedback polynomial of LFSR is known to the attacker, which is a common assumption in cryptanalysis). In particular, within the class of key recovery attacks, we concentrate on *query-based key recovery attacks* (abbreviated as ***"QuBaR attacks"*** in the rest of the paper); QuBaR attacks shall be defined formally in Sec. III-B. The following assumptions are made in this attack model:

A1: The length-M input sequence $\mathbf{x}_1^M$ is assumed to be a realization of an i.i.d. Bernoulli process with parameter $1/2$.

A2: The length-N output sequence $\mathbf{z}_1^N$ is assumed to be given to the attacker, where $N, M \in \mathbb{Z}^+$ (note that, this implies we necessarily have $M > N \geq 1$ due to Definiton 2.1).

A3: Explicit knowledge of the feedback polynomial of the generating LFSR is not used.

A4: The degree of the feedback polynomial of the generating LFSR, i.e., L, is sufficiently large.

Note that assumption A3 will be further clarified after we describe QuBaR attack model precisely. Further, from now on we denote the input sequence as $\mathbf{X}_1^M$ and the corresponding internal state sequence as $\mathbf{Y}_1^M$ due to the stochastic nature of the input and hence the internal state sequences. Next, we continue with the following definitions.

Definition 3.1: The symbol H_i denotes the index of the i-th $\varnothing$ in $\mathbf{Y}_1^M$, for $0 \leq i \leq N$.

Note that, since we have $Y_0 = \varnothing$ with probability 1 by convention, we also use $H_0 = 0$ with probability 1 as the initial condition for $\{H_i\}$.

Definition 3.2: We define $Q_i \stackrel{\triangle}{=} H_i - H_{i-1} - 2$, for $1 \leq i \leq N$.

Remark 3.1: For each Q_i (regardless of its particular realization), the ABSG algorithm generates an output bit z_i. Thus, the number of output bits in the ABSG algorithm is precisely equal to the number of corresponding $\{Q_i\}$.

Next, we state the following result regarding the distribution of $\{Q_i\}$, which will be heavily used throughout the rest of the paper.

Lemma 3.1: Under assumptions A1 and A2, the random variables $\{Q_i\}$ are i.i.d. with geometric p.m.f. of parameter $1/2$:

$$p\left(q_i\right) \stackrel{\triangle}{=} \Pr\left[Q_i = q_i | \mathbf{z}_1^N\right] = (1/2)^{q_i+1}, \text{ for } q_i \in \mathbb{N},\ 1 \leq i \leq N. \tag{1}$$

Proof: See Appendix I. ■

B. Problem Formulation

In this section, we provide an analytical formulation of the problem considered in this paper. As the first step, we show that, under assumptions A1, A2, A3, and A4, all key recovery attacks to ABSG are equivalent to recovering the exact realizations of $\mathbf{Q}_1^N$, stated in Theorem 3.1[2]:

Theorem 3.1: Under the assumptions A1, A2, A3 and A4, the following three computational problems are equivalent in the sense of probabilistic polynomial time reducibility [17]:

1) Retrieving any L independent linear equations in terms of $\mathbf{X}_1^M$.
2) Retrieving any L consecutive bits from $\mathbf{X}_1^M$.
3) Correctly guessing $\mathbf{Q}_i^{\theta+i-1}$ for any positive integers i and θ such that

$$\sum_{j=i}^{\theta+i-1} (q_j + 2) \geq L, \tag{2}$$

is satisfied.

Proof: See Appendix II. ■

Next, we introduce the model for the query type attacks, namely *QuBaR attacks*, which are considered throughout the paper. Qualitatively, a QuBaR attack consists of repeating the following procedure: For a cryptosystem that has a secret, generate a "guess", which aims to guess the secret itself, and subsequently "checks" whether the guess is equal to the secret or not; if the guess is equal to the secret, then terminate the procedure, else continue with another guess. The maximum number of guesses proposed in this procedure are limited by the complexity of the QuBaR attack, which is provided as an input parameter to the attack algorithm. Note that, if the task at hand is to guess i.i.d. random variables (which is the case for the third problem of Theorem 3.1), the QuBaR attack model is intuitively obviously reasonable. Furthermore, recall that most of the cryptanalysis against symmetric key cryptography may be modeled in this way (e.g., time-memory attacks, correlation attacks, algebraic attacks and alike). Next, we formally present the general form of QuBaR attack algorithms.

Definition 3.3: Assuming the existence of a "check algorithm" $\mathcal{T}(G)$ for a "guess" G (the output of $\mathcal{T}(G)$ is 1 if and only if the guess G is equal to the secret), a QuBaR attack algorithm, of complexity $\mathcal{C}$, executes the following steps:

For $k = 1$ to $\mathcal{C}$

1. Generate a guess G_k.
2. Compute $\mathcal{T}(G_k)$.
3. If $\mathcal{T}(G_k) = 1$, then terminate and output the secret given by G_k.

end

Next, we introduce the particular "guess" structure (together with the accompanying relevant definitions) which aims to find $\mathbf{Q}_i^{\theta+i-1}$ so as to solve the third computational problem of Theorem 3.1.

[2]For the random variable Q_i, its realization is denoted by q_i.

Definition 3.4: An *ABSG-guess* is a triplet defined as $G \triangleq \{i, \theta, \mathbf{q}_i^{\theta+i-1}\}$, such that $2\theta + \beta \geq L$, where $\beta \triangleq \sum_{j=0}^{\theta-1} q_{i+j}$, $i \geq 1$ and $i + \theta - 1 \leq N$.

The Bernoulli random variable, $\mathcal{T}(G_k)$, indicates the success probability of guess G_k and is heavily used throughout the rest of the paper, where $G_k \triangleq \left(i_k, \theta_k, \mathbf{q}_{i_k}^{\theta_k+i_k-1}\right)$ is the ABSG-guess of a QuBaR attack (against ABSG) at step k. Note that, at each step k, the "guessed" values $\mathbf{q}_{i_k}^{\theta_k+i_k-1}$ themselves depend on k, which is not explicitly stated (unless otherwise specified) for the sake of notational convenience; this should be self-understood from the context.

Remark 3.2: Note that, the probability of having a successful QuBaR attack *after precisely K steps* is equal to $\Pr[\mathcal{T}(G_1) = 0, \mathcal{T}(G_2) = 0, \ldots \mathcal{T}(G_{K-1}) = 0, \mathcal{T}(G_K) = 1]$ which is *not* equal to $\Pr(\mathcal{T}(G_K) = 1)$ (the latter being equal to the marginal successful guess probability at step K). Moreover, neither of these expressions is the success probability of any QuBaR attack with a specified complexity, which will formally be defined in (4). Observe that our formulation allows the usage of *potentially correlated guesses* $\{G_k\}$ which aims to make the approach as generic as possible.

Corollary 3.1: Per Lemma 3.1 and Definition 3.4, we have

$$\Pr[\mathcal{T}(G_k) = 1] = \Pr\left[\mathbf{Q}_{i_k}^{i_k+\theta_k-1} = \mathbf{q}_{i_k}^{i_k+\theta_k-1} \,|\, \mathbf{z}_1^N\right] = \prod_{j=i_k}^{i_k+\theta_k-1} \left(\frac{1}{2}\right)^{q_j+1} = \left(\frac{1}{2}\right)^{\beta_k+\theta_k}, \tag{3}$$

where $\beta_k \triangleq \sum_{j=0}^{\theta_k-1} q_{i_k+j}$.

The following corollary, which is a direct consequence of Theorem 3.1, is one of the key results of the paper.

Corollary 3.2: All QuBaR-type attacks against ABSG are probabilistic polynomial time reducible to the QuBaR algorithm (defined in Definition 3.3) which uses ABSG-guesses defined in Definition 3.4 and aims to find $\mathbf{Q}_i^{\theta+i-1}$ satisfying (2) for any $i, \theta \in \mathbb{Z}^+$.

Definition 3.5: From now on, we call an arbitrary "ABSG-Guess", G, simply as "guess". Further, for the sake of notational convenience, we use

$$\mathfrak{A} = \{G_k\}_{k=1}^{\mathcal{C}(\mathfrak{A})}$$

for any attack algorithm $\mathfrak{A}$ mentioned in Corollary 3.2, where $\mathcal{C}(\mathfrak{A})$ denotes the (algorithmic) complexity of $\mathfrak{A}$ (i.e., number of guesses applied within $\mathfrak{A}$). Accordingly, the success probability of any $\mathfrak{A}$ is given by

$$\Pr_{succ}(\mathfrak{A}) \triangleq \Pr\left[\vee_{k=1}^{\mathcal{C}(\mathfrak{A})} \mathcal{T}(G_k) = 1\right] = 1 - \Pr\left[\wedge_{k=1}^{\mathcal{C}(\mathfrak{A})} \mathcal{T}(G_k) = 0\right]. \tag{4}$$

Hence, as far as QuBaR attacks against ABSG are concerned, w.l.o.g., in this paper we focus on the ones specified in Corollary 3.2, which aim to solve the third computational problem of Theorem 3.1. In particular, in the rest of the paper, we explore the fundamental limits of the aforementioned QuBaR attacks (denoted by $\mathfrak{A}$) under various setups of interest.

Remark 3.3:

(i) Measure of QuBaR Complexity in Terms of L: At first glance, it may look reasonable to evaluate the complexity of a QuBaR attack in terms of the length of its input, which is N since the input is $\mathbf{z}_1^N$. Note that, this is a common practice in complexity theory. However, when we confine the setup as the application of a QuBaR attack to the ABSG algorithm (prior to which there exists an LFSR whose length-L initial state is unknown), then it would be more reasonable to evaluate the complexity of a QuBaR attack in terms of L (since we eventually aim to find L consecutive bits of $\mathbf{X}_1^M$; see Theorem 3.1). This is precisely the approach we pursue in this paper, i.e., the analysis of the resulting QuBaR attack complexity is given as a function of L.

(ii) Time Complexity of QuBaR: First, note that the time complexity of a QuBaR attack (denoted by $\mathfrak{A}$) is given by the product of $\mathcal{C}(\mathfrak{A})$, the complexity of generating a guess and the complexity of checking a guess. Hence, the quantity $\mathcal{C}(\mathfrak{A})$ forms a lower bound on the time complexity of the QuBaR attack, $\mathfrak{A}$. Furthermore, complexities of both generating a guess and checking a guess may, in practice, be considered to be of $poly(L)$ (see item (v) of this remark). Moreover, we will soon show that at optimality $\mathcal{C}(\mathfrak{A})$ is of $exp(L)$. Hence, at optimality, the lower bound of $\mathcal{C}(\mathfrak{A})$ is, in practice, tight to the first order in the exponent. Therefore, throughout this paper, we "treat" the quantity of $\mathcal{C}(\mathfrak{A})$ as the time complexity of a QuBaR attack $\mathfrak{A}$ and carry out the analysis accordingly.

(iii) Data Complexity of QuBaR: First, note that, for a QuBaR attack $\mathfrak{A}$, consisting of guesses $\{G_k\}$, the data complexity of the k-th guess $G_k = \left(i_k, \theta_k, \mathbf{q}_{i_k}^{\theta_k+i_k-1}\right)$ is, by definition, θ_k. We will soon show that, at "general case" optimality we have $\theta_k = \mathcal{O}(L)$ for each k. Hence, the data complexity of an optimal QuBaR attack $\mathfrak{A}$ is at most $\mathcal{C}(\mathfrak{A}) \cdot \mathcal{O}(L)$. Furthermore, we will show that at optimality $\mathcal{C}(\mathfrak{A})$ is of $exp(L)$. Hence, we conclude that, at optimality $\mathcal{C}(\mathfrak{A})$ is a tight (to the first order in the exponent) upper bound on the data complexity[3].

(iv) Algorithmic Complexity of QuBaR: In parts (ii) and (iii) above, we stress that for an optimal QuBaR attack algorithm $\mathfrak{A}$ against ABSG, $\mathcal{C}(\mathfrak{A})$ forms a *tight* lower (resp. upper) bound on the time (resp. data) complexity of $\mathfrak{A}$, to the first order in the exponent[4]. Following the general convention in cryptanalysis, we use the term "algorithmic complexity" as the maximum of time complexity and data complexity. Thus, we conclude that, at optimality the algorithmic complexity is equal to the time complexity, Furthermore, at optimality, the time complexity, the data complexity and $\mathcal{C}(\mathfrak{A})$ are all equal to each other to the first order in the exponent. Our subsequent developments are based on analytical quantification of $\mathcal{C}(\mathfrak{A})$. Moreover, due to the aforementioned reasons, our results on $\mathcal{C}(\mathfrak{A})$ apply (to the first order in the exponent) to the time complexity, the data complexity and the algorithmic complexity, as well.

(v) Practical Implementation Approaches to QuBaR Algorithms: As far as practical attacks are concerned, existence of a

[3]This result is valid for the general case QuBaR attacks, analyzed in Sec. V. For a restricted class of QuBaR attacks, namely "Exhaustive-Search Type" QuBaR attacks (analyzed in Sec. IV), we show that, at optimality $\mathcal{C}(\mathfrak{A})$ is a loose upper bound on the data complexity.

[4]Once again, the argument in this remark is valid for the general case QuBaR attacks of Sec. V.

polynomial-time guess generation algorithm is obvious. Furthermore, a polynomial-time check algorithm, which corresponds to the procedure of initiating a LFSR (whose feedback polynomial is assumed to be known) with the corresponding "guessed and retrieved" L consecutive bits of $\mathbf{X}_1^M$, generating sufficiently many output bits and comparing them with the original output bits, constitutes a practical approach.

(vi) Relationship Of QuBaR Attacks With State-Of-The-Art Attack Algorithms: We see that QuBaR attacks are analogous to "first type of attacks" described in [15], which "aim to exploit possible weaknesses of compression component introduced by ABSG". However, note that, QuBaR attacks *do not* use explicit knowledge of the feedback polynomial of the generating LFSR, (recall the structure of algorithm $\mathcal{T}$) which is a direct consequence of the assumption A3.

IV. Optimum Exhaustive-Search Type QuBaR Attacks Against ABSG

In this section, we deal with "exhaustive-search" type QuBaR attacks which are formally defined in Definition 4.1. Qualitatively, given the output sequence $\mathbf{z}_1^N$, an exhaustive-search type QuBaR attack aims to correctly identify θ-many $\{Q_i\}$ (equivalently at least L consecutive bits of $\mathbf{X}_1^M$ per Theorem 3.1) beginning from *an arbitrarily-chosen, fixed index*, subject to constraint (2) [5]. Since the attacker is confined to initiate the guesses beginning from a fixed index for exhaustive-search attacks, in practice this can be thought to be equivalent to a scenario where the attacker uses only a *single portion* of the observed output sequence $\mathbf{z}_1^N$.

First theorem of this section, namely Theorem 4.1, proves the existence of an exhaustive-search type QuBaR attack with success probability of $1-\epsilon$ (for any $\epsilon > 0$) with algorithmic complexity $2^{2L/3}$ (in particular, with time complexity $2^{2L/3}$ and data complexity $L/3$) under the assumptions mentioned in Section III-A. The second theorem of this section, namely Theorem 4.2, proves that the algorithmic complexity of the best (in the sense of $\mathcal{C}$) exhaustive-search type QuBaR algorithm under the assumptions A1, A2, A3, A4 is lower-bounded by $2^{2L/3}$ (to the first order in the exponent). Hence, as a result of these two theorems, we show that the overall algorithmic complexity of the best exhaustive-search attack against ABSG has complexity $2^{2L/3}$ to the first order in the exponent (argued in Corollary 4.1). Note that, in [15] Gouget et. al. mention the existence of an exhaustive-search attack (under i.i.d. Bernoulli $1/2$ input assumption) of complexity $\mathcal{O}\left(2^{2L/3}\right)$ without providing the details of the attack. Our main novelty in this section is that, we provide a rigorous proof about the existence of such an attack (Theorem 4.1, which is analogous to the "achievability"-type proofs in traditional lossless source coding) and further show that this is the best (to the first order in the exponent) in the sense of algorithmic complexity under some certain assumptions, specifically within the class of exhaustive-search QuBaR attacks (Theorem 4.2, which is analogous to the "converse"-type proofs in traditional lossless source coding). As a result, the developments in this section can be considered to be analogous to those of source coding by Shannon [18]; see Remark 4.3 for a further discussion on this subject. Theorem 4.3

[5]In contrast with exhaustive-search attacks, we also consider a generalized version, where we focus on identifying θ-many $\{Q_i\}$, possibly beginning from arbitrarily-chosen, multiple indices, which constitutes the topic of Sec. V.

concludes the section, which characterizes some necessary conditions of the optimal exhaustive-search type QuBaR attacks against ABSG.

We begin our developments with the formal definition of exhaustive-search type QuBaR attacks.

Definition 4.1: The class of exhaustive-search type QuBaR attacks against ABSG are defined as

$$\mathcal{S}^E \triangleq \{\mathfrak{A}^E = \{G_k\}_{k=1}^{\mathcal{C}(\mathfrak{A}^E)} : \forall k, i_k = 1\}, \tag{5}$$

where each k-th guess $G_k = \left(i_k, \theta_k, \mathbf{q}_{i_k}^{\theta_k+i_k-1}\right)$ is subject to (2) (see Definition 3.4).

Remark 4.1: Exhaustive-search type attacks constitute an important class of attacks in cryptanalysis. They essentially determine the "effective size" of the key space of any cipher. In case of ABSG, as we mentioned at the beginning of this section, since the exhaustive-search type QuBaR attack uses a single portion of the output sequence, they form a basic choice for practical cryptanalysis via QuBaR attacks in situations where a limited amount ($poly\,(L)$) of output data are available to the attacker.

Thus, at each k-th step, via guess G_k an exhaustive-search type QuBaR attack aims to correctly identify θ_k-many $\{Q_i\}$ subject to (2) beginning from a fixed index i_k, equivalently at least L consecutive bits of $\mathbf{X}_1^M$ beginning from the index i'_k (in general $i'_k \neq i_k$ due to the "decimation" nature of ABSG). As we specified in Definition 4.1, in our developments w.l.o.g. we use $i_k = 1$ (which in turn implies having $i'_k = 1$ as well).

Theorem 4.1: (Achievability - Exhaustive-Search) Under the assumptions A1, A2, A3, A4, mentioned in Section III-A, there exists an exhaustive-search type QuBaR attack algorithm $\mathfrak{A}^E_{ach,opt}$ against ABSG with $\mathcal{C}\left(\mathfrak{A}^E_{ach,opt}\right) = 2^{2L/3}$ such that $\mathrm{Pr}_{succ}\left(\mathfrak{A}^E_{ach,opt}\right) > 1-\epsilon$, for any $\epsilon > 0$. Further, $\mathcal{C}_{ave}\left(\mathfrak{A}^E_{ach,opt}\right) = \frac{1}{2}\left(2^{2L/3}+1\right)$ where $\mathcal{C}_{ave}\left(\mathfrak{A}^E_{ach,opt}\right)$ is the *expected complexity* of $\mathfrak{A}^E_{ach,opt}$ over the probability distribution induced by $\mathbf{q}$.

Proof: See Appendix III. ■

Remark 4.2: An inspection of the proof of Theorem 4.1 reveals that (as promised in Remark 3.3) the overall data complexity of the proposed attack algorithm $\mathfrak{A}^E_{ach,opt}$ is $L/3$ which certainly implies that each guess is of data complexity $\mathcal{O}\,(L)$. Furthermore, the overall time complexity of $\mathfrak{A}^E_{ach,opt}$ is $\mathcal{O}\left(2^{2L/3}\right)$ assuming that the contribution of the generation of each guess is $poly\,(L)$ (which is reasonable in practice). Note that, the time and data complexity of the proposed attack $\mathfrak{A}^E_{ach,opt}$ used in the proof of Theorem 4.1 coincides with the one mentioned in [15].

Next, we prove the converse counterpart of Theorem 4.1, namely derive a lower bound on the algorithmic complexity of any exhaustive-search type QuBaR attack with an inequality constraint on the success probability.

Theorem 4.2: (Converse - Exhaustive-Search) Under the assumptions A1, A2, A3, A4, and for any $\mathfrak{A}^E \in \mathcal{S}^E$ with $\mathrm{Pr}_{succ}\left(\mathfrak{A}^E\right) > \frac{1}{2}$, we necessarily have $\mathcal{C}\left(\mathfrak{A}^E\right) > \underline{\mathcal{C}}^E_{min} \triangleq 2^{2L/3}\left(\frac{1}{2} - \frac{6}{L}\right)$.

Proof: See Appendix IV. ■

Corollary 4.1: After some straightforward algebra, it can be shown that

$$\mathcal{C}\left(\mathfrak{A}_{ach,opt}^{E}\right) \doteq \mathcal{C}_{ave}\left(\mathfrak{A}_{ach,opt}^{E}\right) \doteq \underline{\mathcal{C}}_{min}^{E}$$

in L. Thus, Theorems 4.1 and 4.2 show that, under the assumptions mentioned in Section III-A, the *tight* lower bound (to the first order in the exponent) on the algorithmic complexity of any exhaustive-search type QuBaR attack against ABSG is $2^{2L/3}$.

Following remark provides the promised discussion at the beginning of the section, which interprets the relationship between the result proved in this section (namely, Theorems 4.1 and 4.2) and the traditional lossless source coding of information theory.

Remark 4.3: Observe that for the exhaustive-search setup, the problem is "somewhat dual" of the lossless source coding problem. Intuitively, the concept of cryptographic compression (which is also termed as "decimation" in this paper) aims to produce a sequence of random variables, such that the sequence is as long as possible with the highest entropy possible so as to render cryptographic attacks useless as much as possible (which amounts to making the decimation operation "non-invertible" in practice). On the other hand, in lossless source coding, the goal is to produce an output sequence which is as short as possible while maintaining "exact invertibility" (which amounts to "lossless" decoding). Hence, it is not surprising that, from the cryptanalyst's point of view, usage of concepts from lossless source coding may be valuable. To be more precise, the cryptanalyst aims to identify a set of *highly-probable* sequences (each of which is a collection of i.i.d. random variables from a known distribution), of which cardinality is as small as possible, thereby maximizing the chances of a successful guess with the least number of trials. As a result, the usage of the concept of *typicality* fits naturally within this framework. In particular, typicality is the essence of the proof of the converse theorem (Theorem 4.2), which states a fundamental lower bound on the complexity of all possible exhaustive-search type QuBaR attacks. *The outcome of "converse" states a negative result (which is unknown for the case of stream ciphers to the best of our knowledge) within a reasonable attack class in cryptanalysis by construction*. This observation contributes to a significant portion of our long-term goal, which includes construction of a unified approach to cryptanalysis of stream ciphers. In particular, our future research includes focusing on specific cryptosystems and quantifying fundamental bounds on the performance of attacks (within a pre-specified reasonable class) against these systems.

Following theorem characterizes some important necessary conditions for an optimal exhaustive-search type QuBaR attack against ABSG, subject to an equality constraint on the success probability. Thus, these results are important in practice since they provide some guidelines in construction of optimal or near-optimal exhaustive-search type QuBaR attacks.

Theorem 4.3: Given an optimal (in the sense of minimizing $\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)$ subject to an equality constraint on the success probability) exhaustive-search type QuBaR attack (denoted by $\mathfrak{A}_{opt}^{E}$) against ABSG, we have the following necessary conditions:

(i) The corresponding guesses are *prefix-free*.

(ii) The corresponding "success events" $\{\mathcal{T}(G_i)=1\}_{i=1}^{\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)}$ are *disjoint*.

(iii) We have

$$\Pr{}_{succ}\left(\mathfrak{A}_{opt}^{E}\right)=\Pr\left(\vee_{k=1}^{\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)}\left[\mathcal{T}(G_k)=1\right]\right)=\sum_{k=1}^{\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)}\Pr\left(\mathcal{T}(G_k)=1\right). \tag{6}$$

(iv) The corresponding "success events" $\{\mathcal{T}(G_i)=1\}_{i=1}^{\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)}$ satisfy

$$(i>j)\quad\Longrightarrow\quad\left[\Pr\left(\mathcal{T}(G_i)=1\right)\leq\Pr\left(\mathcal{T}(G_j)=1\right)\right],$$

for any $i\neq j$, such that, $i,j\in\left\{1,\ldots,\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)\right\}$.

Proof: See Appendix V. ■

V. Optimum QuBaR Attacks Against ABSG (General Case)

In this section, we consider the "general case QuBaR attacks", i.e., we relax the condition of being "exhaustive-search", which amounts to relaxing the condition of $i_k=1$ for $\{G_k\}$ in (5). Thus, the goal of the attacker is to guess the true values of $\mathbf{Q}_i^{i-\theta+1}$, subject to (2), for an arbitrary initial index i, equivalently (cf. Theorem 3.1) the attacker's goal is to retrieve *any* (at least) L consecutive bits from the input sequence $\mathbf{X}_1^M$. Note that, this setup implies that exponential amount of output bits are available to the attacker for the cryptanalysis. As we will show in the sequel, via following this formulation, we can improve the time-complexity (and hence the overall algorithmic complexity) at the expense of an exponential increase in the data complexity (which does not affect the overall algorithmic complexity). Thus, the general case can be viewed as one extreme regarding the time-data tradeoff; the other extreme is the exhaustive-search type attacks covered in the previous section.

Similar to the exhaustive-search case, we prove an achievability result first, namely Theorem 5.1, (which is simply the "most-probable choice attack" of [7], [15]), which implies the existence of a QuBaR attack of algorithmic complexity $2^{L/2}$ under the assumptions mentioned in Section III-A. Next, we provide the converse theorem for the general case, which states that the best QuBaR attack's algorithmic complexity is lower bounded with $2^{L/2-1}$ under the assumptions A1, A2, A3, A4. Hence, we conclude that, to the first order in the exponent, the best QuBaR attack against ABSG is of complexity $2^{L/2}$.

We begin our development with the following definition.

Definition 5.1: The set of "successful" QuBaR attacks is defined as

$$\mathcal{S}_p\stackrel{\triangle}{=}\left\{\mathfrak{A}=\{G_k\}_{k=1}^{\mathcal{C}(\mathfrak{A})}:\Pr\left(\vee_{k=1}^{\mathcal{C}(\mathfrak{A})}\left[\mathcal{T}(G_k)=1\right]\right)>1/2\right\}. \tag{7}$$

We treat any $\mathfrak{A}\in\mathcal{S}_p$ as a *successful QuBaR attack* and derive an achievable (to the first order in the exponent) lower bound on the complexity of these attacks.

First, for the sake of completeness, we state the achievability result via providing an extended version of the proof regarding the complexity of the proposed "most probable choice attack" of [7], [15] using our notation.

Theorem 5.1: [7], [15] (Achievability - General Case) Under the assumptions A1, A2, A3, A4, mentioned in Section III-A, there exists a QuBaR attack algorithm $\mathfrak{A}_{ach,opt} \in \mathcal{S}_p$ against ABSG with $\mathcal{C}\left(\mathfrak{A}_{ach,opt}\right) = 2^{L/2}$. Further, $\mathcal{C}_{ave}\left(\mathfrak{A}_{ach,opt}\right) = \frac{1}{2}\left(2^{L/2}+1\right)$ where $\mathcal{C}_{ave}\left(\mathfrak{A}_{ach,opt}\right)$ is the *expected* complexity of $\mathfrak{A}_{ach,opt}$ over the probability distribution induced by $\mathbf{q}$.

Proof: See Appendix VI. ■

Remark 5.1: For the proposed attack $\mathfrak{A}_{ach,opt}$, assuming that the generation of all of the guesses $G\left(\cdot\right)$ and the corresponding check algorithm $\mathcal{T}\left(G\left(\cdot\right)\right)$ are $poly\left(L\right)$, both the time and data complexity of the attack can be shown to be equal to $2^{L/2}$ to the first order in the exponent.

Next, we state the converse theorem, which can be viewed as a fundamental result due to its *negative* nature as far as cryptanalysis concerned, to the best of our knowledge.

Theorem 5.2: (Converse - General Case) Under the assumptions A1, A2, A3, A4, and for any $\mathfrak{A} \in \mathcal{S}_p$, we necessarily have $\mathcal{C}\left(\mathfrak{A}\right) > \underline{\mathcal{C}}_{min} \triangleq 2^{L/2-1}$.

Proof: See Appendix VII. ■

Theorems 5.1 and 5.2 imply the following result:

Corollary 5.1: Under the assumptions A1, A2, A3, A4, the tight (to the first order in the exponent) lower bound on algorithmic complexity of any QuBaR attack against ABSG is $2^{L/2}$:

$$\mathcal{C}\left(\mathfrak{A}_{ach,opt}\right) \doteq \mathcal{C}_{ave}\left(\mathfrak{A}_{ach,opt}\right) \doteq \underline{\mathcal{C}}_{min}.$$

Remark 5.2: A practically useful consequence of Corollary 5.1 is as follows: In order to develop a successful "query-based-recovery" (QuBaR) attack (in the sense of being an element of $\mathcal{S}_p$) of complexity less than $\mathcal{O}\left(2^{L/2}\right)$ (say $poly\left(L\right)$), it is necessary to consider a construction where at least one of the assumptions A1, A2, A3, A4 is relaxed. Recalling these assumptions, it is advisable to concentrate on a setup where the assumptions A1 and/or A3 do not apply; in practice, this may lead to using a deterministic approach [16], where explicit knowledge of the generating LFSR's feedback polynomial is utilized and the input sequence to ABSG, $\mathbf{x}$, is an M-sequence[6].

VI. Conclusion

In this paper, we introduce a novel approach to cryptanalysis. We aim to explore *fundamental performance limits* within a specified class of attacks of interest, targeted towards breaking a particular cryptosystem. As a first step, we illustrate our

[6]For further details on M-sequences, we refer the interested reader to [8].

approach via considering the class of "Query-Based Key-Recovery" (QuBaR) attacks against ABSG, which is an LFSR-based stream cipher constructed via irregular decimation techniques. In order to achieve this task, we rely on the following assumptions (which are quite common in conventional cryptanalysis): The input sequence to ABSG is assumed to be an independent identically distributed Bernoulli process with probability $1/2$; the attacker has access to the output sequence of ABSG; an explicit knowledge of the generating LFSR's feedback polynomial is not used; and the degree of the feedback polynomial (denoted by L) of the generating LFSR is sufficiently large. Using these assumptions, we show that breaking ABSG is equivalent to determine the exact realizations of a sequence of random variables, which are proven to be independent identically distributed with geometric distribution of parameter $1/2$. Next, we investigate two setups of interest. In the first setup, we concentrate on the "Exhaustive-Search Type QuBaR" attacks (which form a subset of general-case QuBaR attacks, such that the starting index of all guesses in any element of this set is constrained to be equal to unity). Here, using notions from information theory (in particular asymptotic equipartition property [18]), we prove that the tight lower bound (to the first order in the exponent) on the algorithmic complexity of any successful Exhaustive-Search Type QuBaR attack is $2^{2L/3}$. In the second setup, we concentrate on the general case QuBaR attacks and follow an analogous development to that of the former setup. In particular, we prove that the tight lower bound (to the first order in the exponent) on the algorithmic complexity of any successful QuBaR attack is $2^{L/2}$. Our results can be viewed as a "negative advice" to the cryptanalyst (contrary to the conventional trend in cryptanalysis, where the general goal is to deduce a "negative design advice" to the cryptosystem designer) in terms of QuBaR attacks against ABSG under the aforementioned assumptions.

APPENDIX I
PROOF OF LEMMA 3.1

First, note that each output bit $Z_i = z_i$ (for $1 \leq i \leq N$) is produced by a *block* of input bits from the input sequence $\mathbf{X}_1^M$. In order to identify the i-th input block that generates Z_i (for $1 \leq i \leq N$), we define

$$\begin{aligned} A_i &\triangleq 1 + \sum_{j=1}^{i-1} [Q_j + 2] = H_{i-1} - H_0 + 1 = H_{i-1} + 1, \\ B_i &\triangleq \sum_{j=1}^{i} [Q_j + 2] = H_i - H_0 = H_i, \end{aligned}$$

where we used $H_0 = 0$ as the initial condition. Hence, we note that the input block $\mathbf{X}_{A_i}^{B_i}$ produces the i-th output bit $Z_i = z_i$ which is given per assumption A2. Further, from the definition of the algorithm $\mathcal{B}$ (see Definition 2.1), we have

$$\Pr\left(X_{A_i+1} = z_i \,|\, Z_i = z_i\right) = 1. \tag{I-1}$$

Next, note that the statement of the lemma is *equivalent to*

$$\Pr\left(\mathbf{Q}_1^N = \mathbf{q}_1^N \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) = \prod_{i=1}^{N} \left[\Pr\left(Q_i = q_i \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right)\right] = \prod_{i=1}^{N} \left(\frac{1}{2}\right)^{q_i+1}. \tag{I-2}$$

Thus, it is necessary and sufficient to show (I-2) to prove Lemma 3.1. In order to show (I-2), we use proof by induction.

- Step 1: We would like to show

$$\Pr\left(Q_1 = q_1 \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) = \left(\frac{1}{2}\right)^{q_1+1}. \tag{I-3}$$

Since the value of Q_1 depends only on the first output bit, we have

$$\Pr\left(Q_1 = q_1 \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) = \Pr\left(Q_1 = q_1 \,|\, Z_1 = z_1\right).$$

Next,

$$\begin{aligned}
\Pr\left(Q_1 = 0 \,|\, Z_1 = z_1\right) &= \Pr\left(X_1 = z_1, X_2 = z_1 \,|\, Z_1 = z_1\right), && \text{(I-4)}\\
&= \Pr\left(X_1 = z_1 \,|\, Z_1 = z_1\right), && \text{(I-5)}\\
&= \frac{1}{2}, && \text{(I-6)}
\end{aligned}$$

where (I-4) follows from the definition of the mapping $\mathcal{M}(\cdot,\cdot)$ (Table I), (I-5) follows from (I-1), (I-6) follows from assumption A1. Also, for $q_1 > 0$,

$$\begin{aligned}
\Pr\left(Q_1 = q_1 \,|\, Z_1 = z_1\right) &= \Pr\left(X_1 = \bar{z}_1, X_2 = z_1, \ldots, X_{q_1+1} = z_1, X_{q_1+2} = \bar{z}_1 \,|\, Z_1 = z_1\right), && \text{(I-7)}\\
&= \Pr\left(X_1 = \bar{z}_1, X_3 = z_1, \ldots, X_{q_1+1} = z_1, X_{q_1+2} = \bar{z}_1 \,|\, Z_1 = z_1\right), && \text{(I-8)}\\
&= \left(\frac{1}{2}\right)^{q_1+1} && \text{(I-9)}
\end{aligned}$$

where (I-7) follows from the definition of the mapping $\mathcal{M}\left(\cdot,\cdot\right)$ (Table I), (I-8) follows from (I-1), (I-9) follows from assumption A1. Combining (I-6) and (I-9), we get (I-3).

- Step 2: We assume that

$$\Pr\left(\mathbf{Q}_1^{n-1} = \mathbf{q}_1^{n-1} \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) = \prod_{i=1}^{n-1}\left[\Pr\left(Q_i = q_i \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right)\right] = \prod_{i=1}^{n-1}\left(\frac{1}{2}\right)^{q_i+1}. \tag{I-10}$$

- Step 3: Given (I-10) we want to show that

$$\Pr\left(\mathbf{Q}_1^{n} = \mathbf{q}_1^{n} \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) = \prod_{i=1}^{n}\left[\Pr\left(Q_i = q_i \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right)\right] = \prod_{i=1}^{n}\left(\frac{1}{2}\right)^{q_i+1}. \tag{I-11}$$

Note that, given (I-10), (I-11) is equivalent to

$$\Pr\left(Q_n = q_n \,|\, \mathbf{Q}_1^{n-1} = \mathbf{q}_1^{n-1}, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) = \Pr\left(Q_n = q_n \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) = \left(\frac{1}{2}\right)^{q_n+1}, \tag{I-12}$$

using Bayes rule. Now,

$$\begin{aligned}
\Pr\left(Q_n = 0 \,|\, \mathbf{Q}_1^{n-1} = \mathbf{q}_1^{n-1}, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) &= \Pr\left(X_{A_n} = z_n, X_{A_n+1} = X_{B_n} = z_n \,|\, \mathbf{Q}_1^{n-1} = \mathbf{q}_1^{n-1}, \mathbf{Z}_1^N = \mathbf{z}_1^N\right), && \text{(I-13)}\\
&= \Pr\left(X_{A_n} = z_n \,|\, \mathbf{Q}_1^{n-1} = \mathbf{q}_1^{n-1}, \mathbf{Z}_1^N = \mathbf{z}_1^N\right), && \text{(I-14)}\\
&= \Pr\left(X_{A_n} = z_n \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) = \Pr\left(Q_n = 0 \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right), && \text{(I-15)}\\
&= \Pr\left(X_{A_n} = z_n \,|\, Z_n = z_n\right) = \frac{1}{2} && \text{(I-16)}
\end{aligned}$$

where (I-13) follows from the definition of the mapping $\mathcal{M}(\cdot,\cdot)$ (Table I), (I-14) follows from (I-1), (I-15) and (I-16) follow from assumption A1 [7]. On the other hand, for $q_n > 0$, we have

$$\begin{aligned}
& \Pr\left(Q_n = q_n \,|\, \mathbf{Q}_1^{n-1} = \mathbf{q}_1^{n-1}, \mathbf{Z}_1^N = \mathbf{z}_1^N\right) \\
= \;& \Pr\left(X_{A_n} = \bar{z}_n, X_{A_n+1} = z_n, \ldots X_{B_n-1} = z_n, X_{B_n} = \bar{z}_n \,|\, \mathbf{Q}_1^{n-1} = \mathbf{q}_1^{n-1}, \mathbf{Z}_1^N = \mathbf{z}_1^N\right), && \text{(I-17)} \\
= \;& \Pr\left(X_{A_n} = \bar{z}_n, X_{A_n+2} = z_n, \ldots X_{B_n-1} = z_n, X_{B_n} = \bar{z}_n \,|\, \mathbf{Q}_1^{n-1} = \mathbf{q}_1^{n-1}, \mathbf{Z}_1^N = \mathbf{z}_1^N\right), && \text{(I-18)} \\
= \;& \Pr\left(X_{A_n} = \bar{z}_n, X_{A_n+2} = z_n, \ldots X_{B_n-1} = z_n, X_{B_n} = \bar{z}_n \,|\, \mathbf{Z}_1^N = \mathbf{z}_1^N\right), && \text{(I-19)} \\
= \;& \Pr\left(X_{A_n} = \bar{z}_n, X_{A_n+2} = z_n, \ldots X_{B_n-1} = z_n, X_{B_n} = \bar{z}_n \,|\, Z_n = z_n\right) = \left(\frac{1}{2}\right)^{q_n+1}, && \text{(I-20)}
\end{aligned}$$

where (I-17) follows from the definition of the mapping $\mathcal{M}(\cdot,\cdot)$ (Table I), (I-18) follows from (I-1), (I-19) and (I-20) follow from assumption A1 (see the discussion in the footnote). Combining (I-15), (I-16), (I-19), (I-20), we get (I-12), and equivalently (I-11), which completes the proof.

□

APPENDIX II
PROOF OF THEOREM 3.1

The equivalence of the first and second problems is shown in [15]. In order to prove the theorem, we proceed with proving the equivalence of the second and third problems.

First, we show that the third problem reduces to the second problem in $poly\,(L)$ time: Since we know $\mathbf{z}_1^N$ and $\mathbf{Q}_i^{\theta+i-1}$ per assumption, we construct L consecutive bits of $\mathbf{X}_1^M$ via using Definition 2.1 in the following way. We are given $\mathbf{Q}_i^{i+\theta-1} = \mathbf{q}_i^{i+\theta-1}$ such that (2) holds. Then, we apply the following algorithm:

1) For each $j = i, i+1, \ldots, \theta+i-1$ do:
   a) If $q_j = 0$, generate $B_j = \{z_j, z_j\}$.
   b) If $q_j > 0$, generate $B_j = \left\{\bar{z}_j, z_j^{q_j}, \bar{z}_j\right\}$
2) Concatenate $\{B_j\}_{j=i}^{\theta+i-1}$ thereby forming the desired $\mathbf{X} = \mathbf{x}$ sequence.

Note that, the condition (2) ensures that the resulting $\mathbf{X} = \mathbf{x}$ sequence $\{B_i, B_{i+1}, \ldots, B_{\theta+i-1}\}$ is of length at least L. Furthermore, from the definition of the ABSG algorithm, the resulting $\mathbf{X} = \mathbf{x}$ sequence is unique and necessarily the correct one. Obviously, this algorithm runs in $poly\,(L)$ time, which completes the proof for this case.

Next, we proceed with showing that the second problem can be reduced to the third problem via an algorithm in probabilistic polynomial time. First, note the following Lemma.

[7] Since $\mathbf{X}$ is an i.i.d. Bernoulli $1/2$ process, the value of $\Pr\left(X_{A_n} = z_n \,|\, Z_n = z_n\right)$ is independent of the particular value of A_n and that is why it is equal to $1/2$.

Lemma II-1: Under the assumptions A1, A2, A3, and A4, for any $n \in \mathbb{Z}^+$, we have

$$\Pr\left[\wedge_{l=0}^{poly(L)} Y_{n+l} \neq \varnothing\right] \leq \epsilon, \tag{II-1}$$

for any $\epsilon > 0$ for L sufficiently large.

Proof: First, under the given assumptions, we note the following fundamental results from [16]:

- For any $n \in \mathbb{Z}^+$,

$$\Pr\left(Y_n = \varnothing\right) = \frac{1}{3} + \frac{2}{3}\left(-\frac{1}{2}\right)^n. \tag{II-2}$$

- $\{Y_n\}$ form a Markovian process with memory-1:

$$\Pr\left(Y_n | \mathbf{Y}_1^{n-1} = \mathbf{y}_1^{n-1}\right) = \Pr\left(Y_n | Y_{n-1} = y_{n-1}\right), \tag{II-3}$$

  for any $n \in \mathbb{Z}^+$.

- For any $n \in \mathbb{Z}^+$,

$$\Pr\left(Y_n \neq \varnothing | Y_{n-1} \neq \varnothing\right) = \frac{1}{2}. \tag{II-4}$$

Hence, for any $\epsilon > 0$ we have

$$\begin{aligned}
\Pr\left[\wedge_{l=0}^{poly(L)} Y_{n+l} \neq \varnothing\right] &= \Pr\left(Y_n \neq \varnothing\right) \cdot \prod_{l=1}^{poly(L)} \Pr\left(Y_{n+l} \neq \varnothing | \wedge_{k=0}^{l-1} Y_{n+k} \neq \varnothing\right), && \text{(II-5)}\\
&= \Pr\left(Y_n \neq \varnothing\right) \cdot \prod_{l=1}^{poly(L)} \Pr\left(Y_{n+l} \neq \varnothing | Y_{n+l-1} \neq \varnothing\right), && \text{(II-6)}\\
&= \left[\frac{2}{3} - \frac{2}{3}\left(-\frac{1}{2}\right)^n\right] \cdot \left(\frac{1}{2}\right)^{poly(L)-1}, && \text{(II-7)}\\
&\leq \epsilon && \text{(II-8)}
\end{aligned}$$

where (II-5) follows from Bayes rule, (II-6) follows from (II-3), (II-7) follows from (II-2) and (II-4), (II-8) follows from the fact that the first term in (II-7) is constant in L and the second term is exponentially decaying in L whence ϵ can be made arbitrarily small for sufficiently large L. ■

Now, since $Y_n \in \{0, 1, \varnothing\}$ (i.e., there are constant possibilities for Y_n), w.l.o.g. we assume that Y_n is known. Since we are also given $\mathbf{X}_{n+1}^{n+L}$ for some $n \in \mathbb{Z}^+$, this also means we know $\mathbf{Y}_n^{n+L}$ (via successively applying $\mathcal{M}\left(Y_{n+l-1}, X_{n+l}\right)$ for $l = 1, 2, \ldots, L$). Next, consider the following situations:

1) $\underline{Y_n = Y_{n+L} = \varnothing}$:

   In this case, w.l.o.g. we choose $h_{i-1} = n$ for some i. Next, let K denote the number of $\varnothing$'s within the sequence $\mathbf{Y}_n^{n+L}$ (which is necessarily ≥ 2 per assumption) and assign $\theta = K - 1$. Next, let h_{i+j-2} denote the index of the j-th $\varnothing$ within the sequence $\mathbf{Y}_n^{n+L}$, where $1 \leq j \leq K = \theta + 1$ (implying $h_{i+K-2} = h_{i+\theta-1} = n + L$). Accordingly, assign

$q_j = h_j - h_{j-1} - 2$ for all $j \in \{i, i+1, \ldots, i+\theta-1\}$. Note that, all these $\{h_j\}$ (equivalently $\{q_j\}$) are known since $\mathbf{Y}_n^{n+L}$ is known. Consequently, this means we have identified $\mathbf{Q}_i^{i+\theta-1} = \mathbf{q}_i^{i+\theta-1}$ such that

$$\sum_{j=i}^{\theta+i-1} (q_j + 2) = \sum_{j=i}^{\theta+i-1} (h_j - h_{j-1}) = h_{\theta+i-1} - h_i = L,$$

satisfying the constraint (2). Further, note that the operations performed within this procedure constitute an algorithm, which is in deterministic polynomial time (implying it is also in probabilistic polynomial time).

2) $Y_n = \varnothing$ and $Y_{n+L} \neq \varnothing$:

In this case, since $Y_{n+L} \neq \varnothing$, we aim to identify some $Y_{n+L+L'} = \varnothing$ for $L' > 0$ with high probability in polynomial time. To achieve this task, we consider the sequence $\{Y_{n+L+k}\}$ for $k > 0$. Now, note that as we increment k, after $poly\,(L)$ steps we necessarily need to come across a $\varnothing$ with high probability (the probability of *not* coming across a $\varnothing$ is exponentially small in L per Lemma II-1). Thus, we have $Y_n = Y_{n+L''} = \varnothing$ where $L'' = L + L' > L$. Next, applying algorithmic steps analogous to the ones in Situation 1 (i.e., beginning from $Y_n = Y_{n+L''} = \varnothing$), we identify $\mathbf{Q}_i^{i+\theta-1} = \mathbf{q}_i^{i+\theta-1}$ such that

$$\sum_{j=i}^{\theta+i-1} (q_j + 2) = \sum_{j=i}^{\theta+i-1} (h_j - h_{j-1}) = h_{\theta+i-1} - h_i = L'' > L,$$

satisfying the constraint (2). Further, note that the operations performed within this procedure constitute an algorithm, which is in probabilistic polynomial time.

3) $Y_n \neq \varnothing$ and $Y_{n+L} = \varnothing$:

Our overall goal is to identify (via using an algorithm, which is in probabilistic polynomial time) $Y_{n+L} = Y_{n+L'''} = \varnothing$ such that $L''' - L > L$. In that case, we would be able to apply algorithmic steps analogous to the ones in Situation 1 (i.e., beginning from $Y_{n+L} = Y_{n+L'''} = \varnothing$) and identify $\mathbf{Q}_i^{i+\theta-1} = \mathbf{q}_i^{i+\theta-1}$ such that

$$\sum_{j=i}^{\theta+i-1} (q_j + 2) = \sum_{j=i}^{\theta+i-1} (h_j - h_{j-1}) = h_{\theta+i-1} - h_i = L''' - L > L,$$

satisfying the constraint (2). Next, we show that, beginning from Y_{n+L}, we are able to find some $Y_{n+L'''} = \varnothing$ such that $L''' > 2L$ via a probabilistic polynomial time algorithm. To see this, first consider the sequence $\{Y_{n+L+k}\}$ for $k > 0$ (as we did in Situation 2). Following Lemma II-1 and using similar arguments to Situation 2, we see that as we increment k by $poly\,(L)$, we necessarily come across a $\varnothing$ with high probability. Next, we apply this step $L/2$ times; at each step, we increment k by $poly\,(L)$ and at each step, we see a $\varnothing$ with probability $1 - \epsilon$ where ϵ is exponentially small in L per Lemma II-1. Thus, as a result of incrementing k by a total of $\frac{L}{2} \cdot poly\,(L)$ (which is again $poly\,(L)$), we observe $L/2$ $\varnothing$'s with sufficiently high probability, which makes this procedure an algorithm in probabilistic polynomial time. On the other hand, observing $L/2$ $\varnothing$'s guarantee us to identify some L''' such that $L''' > 2L$ since the gap between two $\varnothing$'s is at least 2 due to the definition of the ABSG algorithm. As a result, we see that we can identify $Y_{n+L'''} = \varnothing$ such that

$L''' - L > L$ via an algorithm which is in probabilistic polynomial time, which was our initial goal.

4) $Y_n \neq \varnothing$ and $Y_{n+L} \neq \varnothing$:

This is straightforward via applying an approach analogous to the Situation 3 above. Again, we begin from Y_{n+L}, consider the sequence Y_{n+L+k} for $k > 0$, increment k in blocks of length $poly\left(L\right)$; the only difference is that this time we use $\frac{L}{2}+1$ blocks (each of which is $poly\left(L\right)$) instead of $\frac{L}{2}$. As a result, we are guaranteed to identify $Y_{n+L''''} = Y_{n+L'''''} = \varnothing$ such that $L''''' - L'''' > L$ via an algorithm which is in probabilistic polynomial time; the rest is obvious.

Thus, the proof of the (probabilistic polynomial time) reduction of the second problem to the third one is completed. Hence the proof of Theorem 3.1. □

APPENDIX III
PROOF OF THEOREM 4.1

For the sake of clarity, throughout this section we use the notation $G_k\left(i_k, \theta_k, \left(\mathbf{q}_{i_k}^{\theta_k+i_k-1}\right)_k\right)$ (instead of $G_k\left(i_k, \theta_k, \mathbf{q}_{i_k}^{\theta_k+i_k-1}\right)$) to denote a particular guess G_k.

Choosing $n \triangleq L/3$, first we define the typical set $A_\epsilon^{(n)}$ with respect to $p\left(q\right)$ (given by (1)):

$$A_\epsilon^{(n)} \triangleq \left\{\mathbf{q}_1^n \,:\, \left|-\frac{1}{n}\log p\left(\mathbf{q}_1^n\right) - H\left(Q\right)\right| \leq \epsilon\right\}, \tag{III-1}$$

where (using logarithm with base-2)

$$H\left(Q\right) = -\sum_{q=0}^{\infty} p\left(q\right)\log p\left(q\right) = 2.$$

At this point, we also recall two fundamental results regarding typical sets [19]:

$$\left(1-\epsilon\right)2^{n(H(Q)-\epsilon)} \,\leq\, \left|A_\epsilon^{(n)}\right| \,\leq\, 2^{n(H(Q)+\epsilon)} \tag{III-2}$$

$$\Pr\left(\mathbf{q}_1^n \in A_\epsilon^{(n)}\right) > 1-\epsilon \tag{III-3}$$

for any $\epsilon > 0$, for sufficiently large n.

Next, we propose the following construction for the attack $\mathfrak{A}_{ach,opt}^{E}$:

1. Index all $\mathbf{q}_1^n \in A_\epsilon^{(n)}$ and accordingly let $\left(\mathbf{q}_1^n\right)_k$ denote the k-th element where $k \in \left\{1, 2, \ldots, \left|A_\epsilon^{(n)}\right|\right\}$. Let $q_{i,k}$ denote the i-th element of $\left(\mathbf{q}_1^n\right)_k$ for $i \in \{1, 2, \ldots, n\}$.

2. At each k-th step of the QuBaR attack, choose $G_k = \left(i_k = 1, \theta_k = n = \frac{L}{3}, \left(\mathbf{q}_1^n\right)_k\right)$; $k \in \left\{1, 2, \ldots, \left|A_\epsilon^{(n)}\right|\right\}$.

Note that, this attack qualifies as a "QuBaR attack against ABSG" only if all of the aforementioned guesses satisfy the constraint (2). To see that this is satisfied for arbitrarily small ϵ, we observe (noting that $\beta_k = \sum_{i=1}^{n} q_{i,k}$)

$$\left|-\frac{1}{n}\log p\left(\left(\mathbf{q}_1^n\right)_k\right) - H\left(Q\right)\right| = \left|\left(1+\frac{\beta_k}{\theta_k}\right) - 2\right| \leq \epsilon \tag{III-4}$$

where the equality follows from (1), the definition of β_k and using $\theta_k = n$, the inequality follows from (III-1). Furthermore, using $\theta_k = n = L/3$, after straightforward algebra (III-4) can be shown to be equivalent to

$$L\left(1-\frac{\epsilon}{3}\right) \leq 2\theta_k + \beta_k \leq L\left(1+\frac{\epsilon}{3}\right).$$

Since we can choose ϵ arbitrarily small, the aforementioned attack qualifies as a QuBaR attack against ABSG as $\epsilon \to 0$.

Next, (III-3) implies that for large n (equivalently for large L) $\Pr_{succ}\left(\mathfrak{A}^E_{ach,opt}\right) = \Pr\left(\mathbf{q}_1^n \in A_\epsilon^{(n)}\right)$ can be made arbitrarily close to 1 since we can choose ϵ arbitrarily small. Thus, $\Pr_{succ}\left(\mathfrak{A}^E_{ach,opt}\right) \to 1$ as $L \to \infty$ and $\epsilon \to 0$. Furthermore, for large L the algorithmic complexity is at most $\left|A_\epsilon^{(n)}\right|$ which can be made arbitrarily close to $2^{2L/3}$ per (III-2) since $n = L/3$, $H(Q) = 2$ and we can choose ϵ arbitrarily small. Thus, the algorithmic complexity is at most $2^{2L/3}$ as $L \to \infty$, $\epsilon \to 0$. Recalling that for sufficiently small ϵ, all elements of $A_\epsilon^{(n)}$ are equiprobable (since $\beta_k \in \mathbb{N}, \theta_k \in \mathbb{Z}^+$) with probability $2^{-(\theta_k+\beta_k)}\big|_{\theta_k=\beta_k=L/3}$, we immediately see that the expected algorithmic complexity is $\frac{1}{2}\left(2^{2L/3}+1\right)$. Note that in the proposed attack, $i_k = 1$ and $\theta_k = n = L/3$ for all k which implies that the corresponding data complexity is $L/3$. □

APPENDIX IV
PROOF OF THEOREM 4.2

First of all, since L is sufficiently large (per assumption A4), we assume w.l.o.g. L is divisible by 6. Our fundamental goal is to characterize the algorithmic complexity of the optimal attacks subject to a lower bound on the success probability of the attack. Thus, we aim to analytically identify

$$\mathfrak{A}^E_{opt} \stackrel{\triangle}{=} \arg\min_{\mathfrak{A}^E \in \mathcal{S}_p^E} \mathcal{C}\left(\mathfrak{A}^E\right), \tag{IV-1}$$

where

$$\mathcal{S}_p^E \stackrel{\triangle}{=} \left\{\mathfrak{A}^E \ : \ \mathfrak{A}^E \in \mathcal{S}^E \text{ and } \Pr\left(\vee_{k=1}^{\mathcal{C}(\mathfrak{A}^E)}\left[\mathcal{T}\left(G_k\right)=1\right]\right) > \frac{1}{2}\right\} \subseteq \mathcal{S}^E,$$

i.e., $\mathcal{S}_p^E$ is a "probabilistically-constrained" subset of $\mathcal{S}^E$ for which the success probability is strictly bounded away from $1/2$. In our terminology, we denote the quantity of $\Pr\left(\vee_{k=1}^{\mathcal{C}(\mathfrak{A})}\left[\mathcal{T}\left(G_k\right)=1\right]\right)$ as the *success probability of algorithm* $\mathfrak{A}$. Our problem is to characterize

$$\mathcal{C}^E_{min} \stackrel{\triangle}{=} \mathcal{C}\left(\mathfrak{A}^E_{opt}\right),$$

in particular, we aim to achieve this goal via quantifying a lower bound on it.

Our proof approach can be summarized as follows: Since it is not a straightforward task to solve the optimization problem (IV-1), we proceed with a simpler problem. We define a set $\tilde{\mathcal{S}}_p^E$, such that $\mathcal{S}_p^E \subseteq \tilde{\mathcal{S}}_p^E \subseteq \mathcal{S}^E$, and accordingly proceed with minimizing $\mathcal{C}\left(\mathfrak{A}^E\right)$ over all $\mathfrak{A}^E \in \tilde{\mathcal{S}}_p^E$. The set $\tilde{\mathcal{S}}_p^E$ is defined in such a way that minimizing $\mathcal{C}\left(\mathfrak{A}^E\right)$ over this set (i.e., over all $\mathfrak{A}^E \in \tilde{\mathcal{S}}_p^E$) is tractable. At the last step, we conclude the proof via deriving a lower bound on the minimum algorithmic complexity over $\tilde{\mathcal{S}}_p^E$, which also forms a lower bound on $\mathcal{C}^E_{min}$ since $\mathcal{S}_p^E \subseteq \tilde{\mathcal{S}}_p^E$.

We proceed with defining the set

$$\tilde{\mathcal{S}}_p^E \triangleq \left\{ \mathfrak{A}^E \ : \ \mathfrak{A}^E \in \mathcal{S}^E \text{ and } \sum_{k=1}^{\mathcal{C}(\mathfrak{A}^E)} \Pr\left(\mathcal{T}\left(G_k\right)=1\right) > \frac{1}{2} \right\}$$

In our terminology, we denote the quantity of $\sum_{k=1}^{\mathcal{C}(\mathfrak{A})} \Pr\left(\mathcal{T}\left(G_k\right)=1\right)$ as the *cumulative success probability of algorithm* $\mathfrak{A}$. Note that, success probability is always upper-bounded by cumulative success probability for any algorithm $\mathfrak{A}$; i.e., we have

$$\Pr\left(\vee_{k=1}^{\mathcal{C}(\mathfrak{A})}\left[\mathcal{T}\left(G_k\right)=1\right]\right) \leq \sum_{k=1}^{\mathcal{C}(\mathfrak{A})} \Pr\left(\mathcal{T}\left(G_k\right)=1\right)$$

due to the union bound, which implies $\mathcal{S}_p^E \subseteq \tilde{\mathcal{S}}_p^E \subseteq \mathcal{S}^E$. Next, we define the optimization problem (which is "alternate" to (IV-1))

$$\tilde{\mathfrak{A}}_{opt}^E \triangleq \arg\min_{\mathfrak{A}^E \in \tilde{\mathcal{S}}_p^E} \mathcal{C}\left(\mathfrak{A}^E\right), \tag{IV-2}$$

and accordingly

$$\tilde{\mathcal{C}}_{min}^E \triangleq \mathcal{C}\left(\tilde{\mathfrak{A}}_{opt}^E\right).$$

In order to quantify the solution of (IV-2), for the sake of convenience we define

$$\mathcal{G}\left(\theta,\alpha\right) \triangleq \left\{ \mathbf{q}_1^\theta \ : \ \forall i,\, q_i \geq 0,\, \theta \in \mathbb{Z}^+,\, \alpha \in \mathbb{N},\, \sum_{i=1}^{\theta} q_i = \beta = L - 2\theta + \alpha \right\} \tag{IV-3}$$

for any given $\theta \in \mathbb{Z}^+$ and $\alpha \in \mathbb{N}$. Observe that $\{\mathcal{G}\left(\theta,\alpha\right)\}$ are clearly disjoint for different pairs of $\{(\theta,\alpha)\}$. Further, note that, by construction, $\mathbf{q}_1^\theta \in \mathcal{G}\left(\theta,\alpha\right)$ for some $\theta \in \mathbb{Z}^+$, $\alpha \in \mathbb{N}$ implies (2) since $2\theta+\beta = L+\alpha \geq L$; thus, any guess $G = \left(1,\theta,\mathbf{q}_1^\theta\right)$ where $\mathbf{q}_1^\theta \in \mathcal{G}\left(\theta,\alpha\right)$ for some $\theta \in \mathbb{Z}^+$, $\alpha \in \mathbb{N}$ is a valid ABSG-guess. Furthermore, any valid guess G necessarily corresponds to a $\mathbf{q}_1^\theta \in \mathcal{G}\left(\theta,\alpha\right)$ for some unique pair (θ,α). Next, using (3) observe that

$$p\left(\mathbf{q}_1^\theta\right) = 2^{-(\theta+\beta)}\Big|_{\beta=L-2\theta+\alpha} = 2^{-(L-\theta+\alpha)}, \tag{IV-4}$$

for any $\mathbf{q}_1^\theta \in \mathcal{G}\left(\theta,\alpha\right)$; i.e., given a pair (θ,α), all elements of $\mathcal{G}\left(\theta,\alpha\right)$ are equally likely with probability $2^{-(L-\theta+\alpha)}$.

Going back to (IV-2), since we are trying to achieve a cumulative success probability strictly greater than $1/2$ using elements from *disjoint* sets $\{\mathcal{G}\left(\theta,\alpha\right)\}$, the optimal strategy is clearly to use the *sorted* elements $\mathbf{q}_1^\theta \in \mathcal{G}\left(\theta,\alpha\right)$ with respect to their success probabilities, specified in (IV-4) [8]. Thus, algorithmically the optimal solution consists of trying the guess with largest marginal success probability first, and then the most probable guess in the remaining ones, and so on.

Next, we aim to characterize the aforementioned sorting process and analyze the minimum number of elements needed to achieve a cumulative success probability strictly greater than $1/2$. Since all elements of $\mathcal{G}\left(\theta,\alpha\right)$ are equally likely (cf. (IV-4)), the problem of sorting individual sequences reduces to the problem of sorting the sets $\{\mathcal{G}\left(\theta,\alpha\right)\}$ in non-increasing order with

[8]This problem is trivially equivalent to the problem of obtaining a pre-specified amount of cake with minimum number of slices, where the slice sizes are fixed, but not necessarily uniform.

respect to (IV-4). The total number of elements in these sorted sets of $\{\mathcal{G}(\theta,\alpha)\}$ such that the total probability exceeds $1/2$ amounts to the sought result $\tilde{\mathcal{C}}_{min}^{E}$. As a result, we should solve the following sorting problem:

Sorting Problem I: Sort over (θ,α), with respect to the cost function $L-\theta+\alpha$, in non-decreasing order, such that

$$(\theta,\alpha)\in\mathcal{S}_{E,F}\stackrel{\triangle}{=}\left\{(\theta,\alpha)\ :\ \theta\in\mathbb{Z}^{+},\ \alpha\in\mathbb{N},\ \beta=L-2\theta+\alpha\geq 0\right\}. \tag{IV-5}$$

Since this sorting needs to be done over (θ,α), our next task is to characterize the feasible set $\mathcal{S}_{E,F}$ over which the sorting will be carried out.

First of all, notice that from the definition of $\mathcal{G}(\theta,\alpha)$ (cf. (IV-3)), we have

$$2\theta-\alpha\leq L, \tag{IV-6}$$

since $\beta=L-2\theta+\alpha\geq 0$. Next, we define

$$B\stackrel{\triangle}{=}L-\theta+\alpha, \tag{IV-7}$$

as our cost function in the aforementioned Sorting Problem I. Note that, for any $\mathbf{q}\in\mathcal{G}(\theta,\alpha)$, $\Pr(\mathbf{Q}=\mathbf{q})=2^{-(\theta+\beta)}=2^{-B}$; i.e., for any guess $G(i,\theta,\mathbf{q})$, its success probability is equal to 2^{-B} where B is computed via (IV-7) using the corresponding θ and α. This means that, for any given guess $G(\cdot)$, its marginal success probability, $\Pr(\mathcal{T}(G)=1)$ is directly determined by the corresponding value of B.

Next, our goal is to find an alternate re-parameterized expression for (IV-5) in terms of B and L since B is our cost function in Sorting Problem I. Now, using (IV-7) in (IV-6) and noting that $\alpha\in\mathbb{N}$ yields

$$\alpha\in\{0,1,\ldots,2B-L\}. \tag{IV-8}$$

which also implies that $B\geq L/2$ since $\alpha\geq 0$. As a side result, this accordingly implies the following upper bound on the marginal success probability of any valid guess:

$$\Pr[\mathcal{T}(G(i,\theta,\mathbf{q}))=1]=2^{-B}\big|_{B=\theta+\beta}\leq 2^{-L/2}\quad\text{for any } G(i,\theta,\mathbf{q})\in\mathcal{G}(\theta,\alpha)\text{ for some }\alpha\in\mathbb{N}. \tag{IV-9}$$

The result (IV-9) will be useful in proving Theorem 5.2 of Section V.

Next, per (IV-7), each value of α uniquely determines θ in terms of B via

$$\theta=L-B+\alpha. \tag{IV-10}$$

Using (IV-10) in (IV-8), we have

$$\theta\in\{L-B,L-B+1,\ldots,B\}, \tag{IV-11}$$

which also implies that $B \leq L-1$ since $\theta \geq 1$. Combining these observations, we find out the following equivalent expression to (IV-5):

$$(\theta,\alpha) \in \mathcal{S}_{E,F} = \bigcup_{B=\frac{L}{2}}^{L-1} \left\{(L-B,0),(L-B+1,1),(L-B+2,2),\ldots,(B-1,2B-L-1),(B,2B-L)\right\}, \quad \text{(IV-12)}$$

where we effectively did a re-parameterization using B. Note that, this re-parameterization allows us to see that, given a fixed B, all $\{\mathcal{G}(\theta,\alpha)\}$ such that

$$(\theta,\alpha) \in \left\{(L-B,0),(L-B+1,1),(L-B+2,2),\ldots,(B-1,2B-L-1),(B,2B-L)\right\},$$

are equivalent to each other in terms of their success probabilities, 2^{-B}. Using this observation and (IV-12), we conclude that Sorting Problem I is equivalent to to the following one:

Sorting Problem II: Sort over (B,α) with respect to B in non-decreasing order, such that

$$(B,\alpha) \in \left\{(B,\alpha)\ :\ \alpha \in \{0,1,\ldots,2B-L\},\ B \in \{L/2,\ldots,L-1\}\right\}. \quad \text{(IV-13)}$$

Note that, the corresponding values of θ in (IV-13) are given by (IV-10).

Following is one of the solutions to Sorting Problem II:

$$\{(B,\alpha)\} = \left\{(L/2,0),\,(L/2+1,0),\,(L/2+1,1),\,(L/2+1,2),\,(L/2+2,0)\ \ldots,(L-1,L-2)\right\}. \quad \text{(IV-14)}$$

Note that all solutions to Sorting Problem II are equivalent to each other in terms of the resulting complexity. In particular, for a given B, we follow the strategy of varying α in increasing order, beginning from 0, ending in $2B-L$ as illustrated in (IV-14).

Next, we concentrate on the range of $L/2 \leq B < 2L/3$ and analyze the corresponding cumulative success probability (denoted by P_1) of the aforementioned strategy (cf. (IV-14)), i.e.,

$$P_1 \stackrel{\triangle}{=} \sum_{B=\frac{L}{2}}^{\frac{2L}{3}-1} \sum_{\alpha=0}^{2B-L} \Pr\left(\mathcal{G}(\theta,\alpha)|_{\theta=L-B+\alpha}\right). \quad \text{(IV-15)}$$

Next, we derive an upper bound on P_1 which will be used in the subsequent computations.

Lemma IV-1: The cumulative success probability in the range of $L/2 \leq B < 2L/3$ (i.e., P_1) is upper-bounded by

$$P_1 \leq \sum_{\theta=\frac{L}{3}+1}^{\frac{2L}{3}-1} \sum_{\alpha=0}^{\theta-\frac{L}{3}-1} \Pr\left(\mathcal{G}(\theta,\alpha)\right). \quad \text{(IV-16)}$$

Proof: From (IV-15), we see that P_1 is defined in the (B,α) space (where $B = L-\theta+\alpha$), over the set

$$\Lambda \stackrel{\triangle}{=} \left\{(B,\alpha)\ :\ \frac{L}{2} \leq B \leq \frac{2L}{3}-1,\, 0 \leq \alpha \leq 2B-L\right\}, \quad \text{(IV-17)}$$

i.e., $P_1 = \sum_{(B,\alpha)\in\Lambda} \Pr\left(\mathcal{G}\left(\theta,\alpha\right)\right)|_{\theta=L-B+\alpha}$. Next, we proceed with defining a set $\tilde{\Lambda}$. The purpose of using this set is to transform the summation indexes to corresponding (θ,α) for each $(B,\alpha)\in\tilde{\Lambda}$. Now we show that $\tilde{\Lambda}$ is a superset of Λ.This is done in four steps.

1) First, recall that

$$\alpha \geq 0. \tag{IV-18}$$

2) Second, observe that

$$\begin{aligned}\left[B \leq \frac{2L}{3} - 1\right] \quad &\Longrightarrow \quad \left[L - \theta + \alpha \leq \frac{2L}{3} - 1\right] \\ &\Longrightarrow \quad \left[\alpha \leq \theta - \frac{L}{3} - 1\right]\end{aligned} \tag{IV-19}$$

3) Third, note that (IV-19) is equivalent to

$$\theta \geq \frac{L}{3} + \alpha + 1 \tag{IV-20}$$

Using (IV-18) in (IV-20) implies

$$\theta \geq \frac{L}{3} + 1 \tag{IV-21}$$

4) Fourth, using $B \leq \frac{2L}{3} - 1$ in $\alpha \leq 2B - L$ (cf. (IV-17)) implies

$$\alpha \leq \frac{L}{3} - 2. \tag{IV-22}$$

Also, using (IV-7) we have

$$\left[\alpha \leq 2B - L = L - 2\theta + 2\alpha\right] \quad \Longrightarrow \quad \left[\theta \leq \frac{L}{2} + \frac{\alpha}{2}\right]. \tag{IV-23}$$

Using (IV-22) in (IV-23) yields

$$\theta \leq \frac{2L}{3} - 1. \tag{IV-24}$$

Now, defining

$$\tilde{\Lambda} \triangleq \left\{(B,\alpha) \ : \ \frac{L}{3} + 1 \leq \theta \leq \frac{2L}{3} - 1,\, 0 \leq \alpha \leq \theta - \frac{L}{3} - 1, \text{ where } B = L - \theta + \alpha\right\},$$

and using (IV-18), (IV-19), (IV-21), (IV-24), we conclude that $\Lambda \subseteq \tilde{\Lambda}$, which implies (IV-16). ■

Next, we proceed with providing an upper bound on the right hand side of (IV-16), which will be shown to be $\mathcal{O}\left(L^{-1}\right)$, i.e., diminishing in L, the length of the generator polynomial of the LFSR[9]. In order to achieve this task, we heavily use the concept of "typical set" (cf. (III-1)). Note that, using (IV-4) and $H(Q) = 2$, (III-1) can be shown to be equivalent to

$$A_\epsilon^{(\theta)} = \left\{\mathbf{q}_1^\theta \ : \ 1 - \epsilon \leq \frac{\beta}{\theta} \leq 1 + \epsilon\right\}. \tag{IV-25}$$

[9]This result, in turn, implies that an optimal QuBaR attack which uses the solution to the Sorting Problem II for $\theta > L/3$ has a negligible cumulative success probability, i.e., negligible success probability.

In the following lemma, we show that all guesses $\{\mathcal{G}(\theta,\alpha)\}$ included in the summation of the right hand side of (IV-16) are necessarily "atypical" (i.e., belong to the complement of the corresponding typical set).

Lemma IV-2: For any $\theta \in \mathbb{Z}^+$, such that $\theta > L/3$, and for all $\alpha \in \mathbb{N}$, such that $0 \leq \alpha \leq \theta - \frac{L}{3}$, we have $\mathcal{G}(\theta,\alpha) \subseteq [A_\epsilon^{(\theta)}]^{(c)}$ for all $\epsilon \in \left(0, \frac{2}{\theta}\right)$, where $[A_\epsilon^{(\theta)}]^{(c)}$ denotes the complement of the typical set $A_\epsilon^{(\theta)}$.

Proof: First of all, note that (cf. (IV-3)), we have

$$\left[\mathbf{q}_1^\theta \in \mathcal{G}(\theta,\alpha)\right] \Rightarrow \left[\frac{\beta}{\theta} = \left(\frac{L}{\theta} - 2\right) + \frac{\alpha}{\theta}\right]. \tag{IV-26}$$

Hence, for any $\mathbf{q}_1^\theta \in \mathcal{G}(\theta,\alpha)$ such that $\theta > L/3$ and $0 \leq \alpha \leq \theta - \frac{L}{3}$, we have

$$\begin{aligned} -\frac{1}{\theta}\log p\left(\mathbf{q}_1^\theta\right) - H(Q) &= \frac{\theta+\beta}{\theta} - 2, && \text{(IV-27)}\\ &= \frac{L-\theta+\alpha}{\theta} - 2 = \frac{L+\alpha}{\theta} - 3, && \text{(IV-28)}\\ &\leq \frac{2L}{3\theta} - 2, && \text{(IV-29)}\\ &\leq \frac{2L}{L+3} - 2 = -\frac{6}{L+3} < 0, && \text{(IV-30)} \end{aligned}$$

where (IV-27) follows from the fact that $p\left(\mathbf{q}_1^\theta\right) = 2^{-(\theta+\beta)}$ and $H(Q) = 2$, (IV-28) follows using (IV-26) in (IV-27), (IV-29) follows since $\alpha \leq \theta - L/3$, (IV-30) follows since $\theta \geq \frac{L}{3} + 1$. Note that (IV-30) implies

$$\left|-\frac{1}{\theta}\log p\left(\mathbf{q}_1^\theta\right) - H(Q)\right| \geq \frac{6}{L+3}. \tag{IV-31}$$

Now, since $\theta \geq \frac{L}{3} + 1$ (equivalently $\frac{2}{\theta} \leq \frac{6}{L+3}$), we have $\epsilon < \frac{6}{L+3}$ for all $\epsilon \in \left(0, \frac{2}{\theta}\right)$. Using this in (IV-31), the claim follows.

■

Next, we provide an upper bound on the right hand side of (IV-16) using Lemma IV-2. For all $\epsilon_\theta \in \left(0, \frac{2}{\theta}\right)$, we have

$$\begin{aligned} \sum_{\theta=\frac{L}{3}+1}^{\frac{2L}{3}-1} \sum_{\alpha=0}^{\theta-\frac{L}{3}-1} \Pr\left(\mathcal{G}(\theta,\alpha)\right) &\leq \sum_{\theta=\frac{L}{3}+1}^{\frac{2L}{3}-1} \Pr\left(\left[A_{\epsilon_\theta}^\theta\right]^{(c)}\right), && \text{(IV-32)}\\ &\leq \sum_{\theta=\frac{L}{3}+1}^{\frac{2L}{3}-1} \epsilon_\theta, && \text{(IV-33)}\\ &\leq \left(\frac{L}{3} - 1\right)\left(\max_{\frac{L}{3}+1\leq\theta\leq\frac{2L}{3}-1} \epsilon_\theta\right), && \text{(IV-34)} \end{aligned}$$

where (IV-32) follows from Lemma IV-2 and the fact that, for any given θ, $\{\mathcal{G}(\theta,\alpha)\}$ are disjoint by construction, (IV-33) follows from (III-3). Now, choosing $\epsilon_\theta = \frac{1}{\theta^2}$ for all θ, and using (IV-34) in (IV-16), we have

$$P_1 \leq \left(\frac{L}{3} - 1\right)\left(\max_{\frac{L}{3}+1\leq\theta\leq\frac{2L}{3}-1} \frac{1}{\theta^2}\right) = \frac{L/3-1}{(L/3+1)^2} < \frac{3}{L}. \tag{IV-35}$$

Thus, for any $\delta_1 > 0$, there exists a sufficiently large L (per assumption A4), where

$$P_1 < \delta_1. \tag{IV-36}$$

Note that, for the optimal strategy, which uses the ordering mentioned in (IV-14), (IV-35) and (IV-36) imply that the range of $\frac{L}{2} \leq B \leq \frac{2L}{3} - 1$ is not sufficient to achieve every given cumulative success probability strictly greater than $1/2$, since δ_1 can be made arbitrarily small. Therefore, we necessarily need to include guesses with $B = 2L/3$ in the optimal structure to achieve a cumulative success probability strictly greater than $1/2$.

Next, we proceed with quantifying the contribution to the cumulative success probability for the case of $B = 2L/3$. In this case, for the optimal strategy, since $\theta = L - B + \alpha$ and $0 \leq \alpha \leq 2B - L$ for a given value of B, the corresponding (θ, α) pairs are of the form $\left\{\left(\frac{L}{3} + \alpha, \alpha\right)\right\}_{0 \leq \alpha \leq L/3}$. Thus, for the case of $B = 2L/3$, the total contribution to the cumulative success probability is given by

$$\Pr\left(\mathcal{G}(L/3, 0)\right) + \sum_{\alpha=1}^{L/3} \Pr\left(\mathcal{G}(L/3 + \alpha, \alpha)\right). \tag{IV-37}$$

Note that, the right hand side of (IV-37) is "atypical" per Lemma IV-2; accordingly, we will show that the only significant contribution to the cumulative success probability is due to the left hand side of (IV-37) since it includes terms within the corresponding typical set.

Next, we provide an upper bound on the right hand side of (IV-37). Defining $P_2 \triangleq \sum_{\alpha=1}^{L/3} \Pr\left(\mathcal{G}(L/3 + \alpha, \alpha)\right)$, for all $\epsilon_\theta \in \left(0, \frac{2}{\theta}\right)$, we have

$$\begin{aligned} P_2 &= \sum_{\theta=\frac{L}{3}+1}^{2L/3} \Pr\left(\mathcal{G}(\theta, \theta - L/3)\right), && \text{(IV-38)} \\ &\leq \sum_{\theta=\frac{L}{3}+1}^{2L/3} \Pr\left(\left[A_{\epsilon_\theta}^{\theta}\right]^c\right), && \text{(IV-39)} \\ &\leq \left(\frac{L}{3}\right)\left(\max_{L/3+1 \leq \theta \leq 2L/3} \epsilon_\theta\right), && \text{(IV-40)} \end{aligned}$$

where (IV-38) follows from using $\theta = (L - B + \alpha)|_{B=2L/3}$, (IV-39) follows from Lemma IV-2, (IV-40) follows using (III-3). Choosing $\epsilon_\theta = \frac{1}{\theta^2}$ for all θ in (IV-40), we have

$$P_2 \leq \frac{L/3}{(L/3+1)^2} < \frac{3}{L}. \tag{IV-41}$$

Thus, for any $\delta_2 > 0$, there exists a sufficiently large L (per assumption A4), where

$$P_2 < \delta_2. \tag{IV-42}$$

Since δ_1 (resp. δ_2) in (IV-36) (resp. (IV-42)) can be made arbitrarily small, we necessarily need to use guesses from the set $\mathcal{G}\left(\frac{L}{3}, 0\right)$ in order to achieve a cumulative success probability strictly greater than $1/2$.

Next, consider the case of $(\theta, \alpha) = \left(\frac{L}{3}, 0\right)$: Note that, for any $\mathbf{q}_1^{L/3} \in \mathcal{G}\left(\frac{L}{3}, 0\right)$, we have

$$p\left(\mathbf{q}_1^{L/3}\right) = 2^{-(2L/3)}. \tag{IV-43}$$

Per (IV-25), (IV-43) implies that $\mathcal{G}\left(\frac{L}{3},0\right) \subseteq A_{\epsilon}^{(L/3)}$ for any $\epsilon > 0$. Furthermore, after some straightforward algebraic manipulations, it can be shown that, for $0 < \epsilon < \frac{3}{L}$, we have $A_{\epsilon}^{(L/3)} \subseteq \mathcal{G}\left(\frac{L}{3},0\right)$; therefore we have

$$\mathcal{G}\left(\frac{L}{3},0\right) = A_{\epsilon}^{(L/3)} \text{ for } 0 < \epsilon < \frac{3}{L}. \tag{IV-44}$$

In fact, (IV-44) constitutes the fundamental crux of the converse proof. This observation implies that, using sufficiently many guesses from the set $\mathcal{G}\left(\frac{L}{3},0\right)$ is both necessary (since δ_1 and δ_2 may be arbitrarily small) and sufficient (since for $0 < \epsilon < \frac{3}{L}$, we have $\Pr\left(\mathcal{G}\left(\frac{L}{3},0\right)\right) = \Pr\left(A_{\epsilon}^{(L/3)}\right) > 1-\epsilon$) to achieve a cumulative success probability strictly greater than $1/2$ for large L (per Assumption A4).

Now, let

$$P_1 + P_2 + P_3 > 1/2, \tag{IV-45}$$

denote the cumulative success probability of optimal attack in the set $\tilde{\mathcal{S}}_p^E$, where P_3 denotes the contribution to the cumulative success probability by the guesses from $\mathcal{G}\left(\frac{L}{3},0\right)$[10]. Using (IV-35) and (IV-41) in (IV-45), we have

$$P_3 > \frac{1}{2} - \frac{6}{L}. \tag{IV-46}$$

Next, let $\mathcal{C}'$ denote the number of sequences used from the set $\mathcal{G}\left(\frac{L}{3},0\right)$. Using (IV-43), we have

$$\mathcal{C}' = P_3/2^{-2L/3}. \tag{IV-47}$$

Combining (IV-46) and (IV-47) yields

$$\left[\mathcal{C}' > 2^{2L/3}\left(\frac{1}{2} - \frac{6}{L}\right)\right] \implies \left[\mathcal{C}\left(\tilde{\mathfrak{A}}_{opt}^E\right) > 2^{2L/3}\left(\frac{1}{2} - \frac{6}{L}\right)\right],$$

since $\mathcal{C}\left(\tilde{\mathfrak{A}}_{opt}^E\right) > \mathcal{C}'$. Next, using $\mathcal{S}_p^E \subseteq \tilde{\mathcal{S}}_p^E$ yields

$$\mathcal{C}_{min}^E = \mathcal{C}\left(\mathfrak{A}_{opt}^E\right) \geq \tilde{\mathcal{C}}_{min}^E = \mathcal{C}\left(\tilde{\mathfrak{A}}_{opt}^E\right) > 2^{2L/3}\left(\frac{1}{2} - \frac{6}{L}\right), \tag{IV-48}$$

where $\mathfrak{A}_{opt}^E$ and $\tilde{\mathfrak{A}}_{opt}^E$ have been defined in (IV-1) and (IV-8), respectively. Hence, the claim finally follows. □

APPENDIX V
PROOF OF THEOREM 4.3

For the sake of clarity, we use the notation $G_k\left(i_k = 1, \theta_k, \left(\mathbf{q}_{i_k}^{\theta_k+i_k-1}\right)_k\right)$ (instead of $G_k\left(i_k = 1, \theta_k, \mathbf{q}_{i_k}^{\theta_k+i_k-1}\right)$) throughout the proof in this section.

(i) First of all, note that letting $\mathfrak{A}_{opt}^E = \{G_k\}_{k=1}^{\mathcal{C}\left(\mathfrak{A}_{opt}^E\right)}$ denote the optimal exhaustive-search type QuBaR attack against ABSG with success probability $\Pr_{succ}\left(\mathfrak{A}_{opt}^E\right)$, the claim is equivalent to the following statement: For any $i \neq j; i,j \in$

[10]Note that, w.l.o.g. we assume that, at step $B = 2L/3$, the proposed optimal attack uses guesses from the set $\mathcal{G}\left(\frac{L}{3},0\right)$ *in the end* (i.e., after applying guesses from the sets $\left\{\mathcal{G}\left(\frac{L}{3}+\alpha,\alpha\right)\right\}_{\alpha=1}^{L/3}$ of which contributions to the cumulative success probability is denoted by P_2). Since our strategy is to "lower-bound" the number of guesses from the set $\mathcal{G}\left(\frac{L}{3},0\right)$ and declare the resulting value as a lower bound on the overall complexity, $\tilde{\mathcal{C}}_{min}$, this approach maintains the validity of our result.

$\{1,\ldots,\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)\}$ (assuming $\theta_j > \theta_i$ w.l.o.g.), we have $\left(\mathbf{q}_1^{\theta_i}\right)_i \neq \left(\mathbf{q}_1^{\theta_i}\right)_j$. Suppose to the contrary, we have $\left(\mathbf{q}_1^{\theta_i}\right)_i = \left(\mathbf{q}_1^{\theta_i}\right)_j$ for some $i \neq j; i,j \in \{1,\ldots,\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)\}$ where w.l.o.g. $\theta_j > \theta_i$. Given $\mathfrak{A}_{opt}^{E}$, we construct an exhaustive-search type QuBaR attack $\tilde{\mathfrak{A}}^{E}$ via eliminating G_j from $\mathfrak{A}_{opt}^{E}$, i.e., $\tilde{\mathfrak{A}}^{E} \triangleq \{\tilde{G}_k\}_{k=1}^{\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)-1}$ where $\tilde{G}_k = G_k$ for $k \in \{1,\ldots,j-1\}$ and $\tilde{G}_k = G_{k+1}$ for $k \in \{j,\ldots,\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)-1\}$. Next, note that

$$[(\mathcal{T}\left(G_j\right)=1) \;\Rightarrow\; (\mathcal{T}\left(G_i\right)=1)] \quad \Longrightarrow \quad \left[\Pr\left(\vee_{k=1}^{\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)}\left[\mathcal{T}\left(G_k\right)=1\right]\right) = \Pr\left(\vee_{1\leq k\leq \mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right),\, k\neq j}\left[\mathcal{T}\left(G_k\right)=1\right]\right)\right]. \tag{V-1}$$

If G_j is a correct guess, then all $\left(\mathbf{q}_1^{\theta_j}\right)_j$ are correct, which implies $\left(\mathbf{q}_1^{\theta_i}\right)_j$ are necessarily correct as well since $\theta_i < \theta_j$. Further, this implies that $\left(\mathbf{q}_1^{\theta_i}\right)_i$ are correct as well per the contradiction assumption. Hence, this proves the left hand side of (V-1); thus, the right hand side of (V-1) is true as well. This, in turn, is equivalent to $\mathrm{Pr}_{succ}\left(\mathfrak{A}_{opt}^{E}\right) = \mathrm{Pr}_{succ}(\tilde{\mathfrak{A}}^{E})$ which yields the promised contradiction ($\mathcal{C}\left(\tilde{\mathfrak{A}}^{E}\right) = \mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right) - 1$) since $\mathfrak{A}_{opt}^{E}$ is an optimal exhaustive-search type QuBaR attack for the given success probability $\mathrm{Pr}_{succ}\left(\mathfrak{A}_{opt}^{E}\right)$; hence the proof the first statement of Theorem 4.3.

(ii) Suppose not; then this means that there exists some $i,j \in \left\{1,2,\ldots,\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)\right\}$, $i \neq j$, such that $\Pr\left[(\mathcal{T}\left(G_i\right)=1) \cap (\mathcal{T}\left(G_j\right)=1)\right] > 0$. This implies that there is some realization $\tilde{\mathbf{q}}$ of $\mathbf{Q}$ with non-zero probability such that the events of $(\mathcal{T}\left(G_i\right)=1)$ and $(\mathcal{T}\left(G_j\right)=1)$ both occur at the same time. In other words, there exists some $\tilde{\mathbf{q}}$ with $\Pr\left(\mathbf{Q}=\tilde{\mathbf{q}}\right) > 0$ such that $\left(\mathbf{q}_1^{\theta_i}\right)_i = \tilde{\mathbf{q}}_1^{\theta_i}$ and $\left(\mathbf{q}_1^{\theta_j}\right)_j = \tilde{\mathbf{q}}_1^{\theta_j}$. However, this implies that $\left(\mathbf{q}_1^{\theta_i}\right)_i$ is a prefix of $\left(\mathbf{q}_1^{\theta_j}\right)_j$ (assuming w.l.o.g. $\theta_i < \theta_j$). Hence contradiction (per the first statement of Theorem 4.3) and the proof of the second statement of Theorem 4.3.

(iii) This statement is the direct consequence of the first and second statements of the theorem.

(iv) First recall that, at optimality $\mathcal{C}\left(\mathfrak{A}_{opt}^{E}\right)$ is the smallest possible value (given the success probability $\mathrm{Pr}_{succ}\left(\mathfrak{A}_{opt}^{E}\right)$). This observation and (6) clearly imply that the optimal strategy consists of "sorted" guesses (in descending order) with respect to the probabilities $\{\Pr\left(\mathcal{T}\left(G_k\right)=1\right)\}$ of the corresponding success events $\{(\mathcal{T}\left(G_k\right)=1)\}$ since the success probability $\mathrm{Pr}_{succ}\left(\mathfrak{A}_{opt}^{E}\right)$ is fixed.

□

APPENDIX VI
PROOF OF THEOREM 5.1

The attack mentioned in the statement of the theorem is the "most probable case attack" given in [7], [15], which consists of simply "trying" a guess of the all zero sequence of $\{Q_i\}$ (of length $L/2$) for non-overlapping windows of output; here we assume w.l.o.g. that L is sufficiently large and even per assumption A4 of Sec. III-A. Formally, this attack can be defined as follows:

$$\mathfrak{A}_{ach,opt} = \{G_k\}_{k=1}^{\mathcal{C}(\mathfrak{A}_{ach,opt})}, \text{ s.t. for each guess } G_k = \left(i_k, \theta_k, \mathbf{q}_{i_k}^{i_k+\theta_k-1}\right),\, i_k = (k-1)\frac{L}{2}+1,\, \theta_k = \frac{L}{2},\, \beta_k = 0,$$

where we recall that, for each k, $\beta_k = \sum_{j=0}^{\theta_k - 1} q_{i_k+j}$ and the probability of G_k's being correct is $\Pr\left[\mathcal{T}\left(G_k\right) = 1\right] = 2^{-(\theta_k+\beta_k)}$. Since $\beta_k = 0$ and $\theta_k = \frac{L}{2}$ for each guess G_k of the proposed attack, we have,

$$\Pr\left[\mathcal{T}\left(G_k\right) = 1\right] = 2^{-L/2}, \quad 1 \leq k \leq \mathcal{C}\left(\mathfrak{A}_{ach,opt}\right). \tag{VI-1}$$

Hence, we have

$$\begin{aligned}
\mathrm{Pr}_{succ}(\mathfrak{A}_{ach,opt}) &= \Pr\left(\vee_{k=1}^{\mathcal{C}(\mathfrak{A}_{ach,opt})}\left[\mathcal{T}\left(G_k\right) = 1\right]\right), \\
&= 1 - \Pr\left(\wedge_{k=1}^{\mathcal{C}(\mathfrak{A}_{ach,opt})}\left[\mathcal{T}\left(G_k\right) = 0\right]\right), \\
&= 1 - \prod_{k=1}^{\mathcal{C}(\mathfrak{A}_{ach,opt})} \Pr\left[\mathcal{T}\left(G_k\right) = 0\right], \qquad \text{(VI-2)} \\
&= 1 - \left(1 - 2^{-L/2}\right)^{\mathcal{C}(\mathfrak{A}_{ach,opt})}, \qquad \text{(VI-3)}
\end{aligned}$$

where (VI-2) follows from the fact that the events of $\{\mathcal{T}\left(G_k\right) = 0\}$ are independent (since they correspond to sequences of non-overlapping windows of $\{Q_i\}$, which are i.i.d.) and (VI-3) follows from (VI-1). Now, recall that

$$\lim_{x \to 0} (1-x)^{1/x} = 1/e \tag{VI-4}$$

Next, choosing $\mathcal{C}\left(\mathfrak{A}_{ach,opt}\right) = 2^{L/2}$, for large L we have

$$\lim_{L\to\infty} \mathrm{Pr}_{succ}(\mathfrak{A}_{ach,opt}) = \lim_{L\to\infty}\left[1 - \left(1 - 2^{-L/2}\right)^{\left(2^{L/2}\right)}\right] = 1 - \frac{1}{e} > \frac{1}{2},$$

which follows from (VI-4). This implies that $\mathfrak{A}_{ach,opt} \in \mathcal{S}_p$, where $\mathcal{C}\left(\mathfrak{A}_{ach,opt}\right) = 2^{L/2}$ for sufficiently large L (per assumption A4). Furthermore, note that all guesses $\{G_k\}$ of the proposed attack $\mathfrak{A}_{ach,opt}$ are equally-likely to succeed (cf. (VI-1)), which subsequently implies that $\mathcal{C}_{ave}\left(\mathfrak{A}_{ach,opt}\right) = \frac{1}{2}\left(2^{L/2}+1\right)$. Hence the proof. □

APPENDIX VII
PROOF OF THEOREM 5.2

Throughout the proof, we assume w.l.o.g. L is even, since it is sufficiently large per assumption A4. We first recall that we have

$$\forall\, k \in \mathbb{Z}^+, \quad \Pr\left(\mathcal{T}\left(G_k\right) = 1\right) \leq \left(\frac{1}{2}\right)^{L/2}, \tag{VII-1}$$

due to (IV-9) of Appendix IV [11].

Next, we proceed with a similar approach to the one pursued in the proof of Theorem 4.2. In particular, we begin with defining a set $\tilde{\mathcal{S}}_p$, which is a superset of $\mathcal{S}_p$, the set of successful QuBaR attacks (cf. (7)):

$$\tilde{\mathcal{S}}_p \triangleq \left\{\mathfrak{A} = \{G_k\}_{k=1}^{\mathcal{C}(\mathfrak{A})} : \sum_{k=1}^{\mathcal{C}(\mathfrak{A})} \Pr[\mathcal{T}\left(G_k\right) = 1] > 1/2\right\}. \tag{VII-2}$$

[11] Note that, in Appendix IV, we derived (IV-9) for exhaustive-search attacks, for which the starting index of the attack is set to unity (cf. (IV-3)). However, after some straightforward algebra, it can be shown that, following (IV-3), all the subsequent derivations of Appendix IV, regarding the "valid ranges of fundamental system parameters", θ, β, α and B, (including the utilized result (IV-9)) are still valid even if we relax the aforementioned condition on the starting index, which amounts to the general case attacks. Thus, (IV-9) can be shown to hold in the case of general QuBaR attacks.

Using the union bound yields

$$\Pr\left(\vee_{k=1}^{\mathcal{C}(\mathfrak{A})}\left[\mathcal{T}\left(G_k\right)=1\right]\right) \leq \sum_{k=1}^{\mathcal{C}(\mathfrak{A})} \Pr\left(\mathcal{T}\left(G_k\right)=1\right).$$

Thus, we have

$$\left[\mathfrak{A} \in \mathcal{S}_p\right] \quad \Longrightarrow \quad \left[\mathfrak{A} \in \tilde{\mathcal{S}}_p\right],$$

which implies

$$\mathcal{S}_p \subseteq \tilde{\mathcal{S}}_p. \tag{VII-3}$$

Further, for any $\mathfrak{A} \in \tilde{\mathcal{S}}_p$, we have

$$\frac{1}{2} < \sum_{k=1}^{\mathcal{C}(\mathfrak{A})} \Pr\left[\mathcal{T}\left(G_k\right)=1\right] \leq \sum_{k=1}^{\mathcal{C}(\mathfrak{A})} (1/2)^{L/2} = 2^{-L/2}\mathcal{C}\left(\mathfrak{A}\right), \tag{VII-4}$$

where the first and the second inequalities follow from (VII-2) and (VII-1), respectively. As a result, we have

$$\min_{\mathfrak{A}\in\mathcal{S}_p} \mathcal{C}\left(\mathfrak{A}\right) \geq \min_{\mathfrak{A}\in\tilde{\mathcal{S}}_p} \mathcal{C}\left(\mathfrak{A}\right) > 2^{L/2-1},$$

where the first inequality follows from (VII-3) and the second inequality follows from the fact that (VII-4) holds for any $\mathfrak{A} \in \tilde{\mathcal{S}}_p$. Hence the proof. □

ACKNOWLEDGEMENT

Authors wish to thank Nafiz Polat Ayerden and Mustafa Orhan Dirik of Boğaziçi University, Turkey for various helpful discussions and comments.

REFERENCES

[1] M. Hellman, "A cryptanalytic time-memory trade-off," *IEEE Trans. Inf. Theory*, vol. IT–26, no. 4, pp. 401–406, Jul. 1980.
[2] A. Biryukov and A. Shamir, "Cryptanalytic time/memory/data tradeoffs for stream ciphers," *Asiacrypt 2000*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 2000, vol. 1976, pp. 1–13.
[3] T. Siegenthaler, "Decrypting a Class of Stream Ciphers Using Ciphertext Only", *IEEE Trans. Comput.*, vol. 34, no. 1, pp. 81–85, Jan. 1985.
[4] W. Meier and O. Staffelbach, "Fast Correlation Attacks on Certain Stream Ciphers", *Journal of Cryptology*, vol. 1, pp. 159–176, 1989.
[5] N. Courtois and W. Meier, "Algebraic Attacks on Stream Ciphers with Linear Feedback", *Advances in Cryptology– EUROCRYPT 2003*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 2003, vol. 2656, pp. 345–359.
[6] N. Courtois, "Fast Algebraic Attacks on Stream Ciphers with Linear Feedback", *Advances in Cryptology– CRYPTO 2003*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 2003, vol. 2729, pp. 177–194, 2003.
[7] A. Gouget, H. Sibert, C. Berbain, N. Courtois, B. Debraize and C. Mitchell, "Analysis of the Bit-Search Generator and Sequence Compression Techniques" in *Fast Software Encryption–FSE 2005*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 2005, vol. 3557, pp. 196–214.
[8] S. W. Golomb, *Shift Register Sequences*, Revised Edition, Aegean Park Press, 1982.
[9] A. Gouget and H. Sibert, "The Bit-Search Generator" in *The State of the Art of Stream Ciphers: Workshop Record, Brugge, Belgium, October 2004*, pp. 60–68, 2004.
[10] D. Coppersmith, H. Krawczyk and Y. Mansour, "The Shrinking Generator" in *Advances in Cryptology–CRYPTO'93*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1993, vol. 773, pp. 22–39.
[11] W. Meier and O. Staffelbach, "The Self-Shrinking Generator" in *Advances in Cryptology– EUROCRYPT'94*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1994, vol. 905, pp. 205–214.
[12] M. Mihaljevic, "A Faster Cryptanalysis of the Self-Shrinking Generator" in *First Australasian Conference on Information Security and Privacy ACISP'96*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1996, vol. 1172, pp. 182–189.
[13] E. Zenner, M. Krause, and S. Lucks, "Improved Cryptanalysis of the Self-Shrinking Generator" in *Australasian Conference on Information Security and Privacy ACISP'01*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 2001, vol. 2119, pp. 21–35.
[14] M. Hell and T. Johansson, "Two New Attacks on the Self-Shrinking Generator," *IEEE Trans. Inf. Theory*, vol. IT–52, no. 8, pp. 3837–3843, Aug. 2006.
[15] A. Gouget and H. Sibert, "How to Strengthen Pseudo-random Generators by Using Compression," in *Advances in Cryptology–EUROCRYPT 2006*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 2006, vol. 4004, pp. 129–146.
[16] Y. Altuğ, N. P. Ayerden, M. K. Mıhçak and E. Anarım, "A Note on the Periodicity and the Output Rate of Bit Search Type Generators," *IEEE Trans. Inf. Theory*, vol. IT–54, no. 2, pp. 666–679, Feb. 2008.
[17] O. Goldreich, *Foundations of Cryptography, Volume 1*, Cambridge University Press, 2001.
[18] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.
[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.